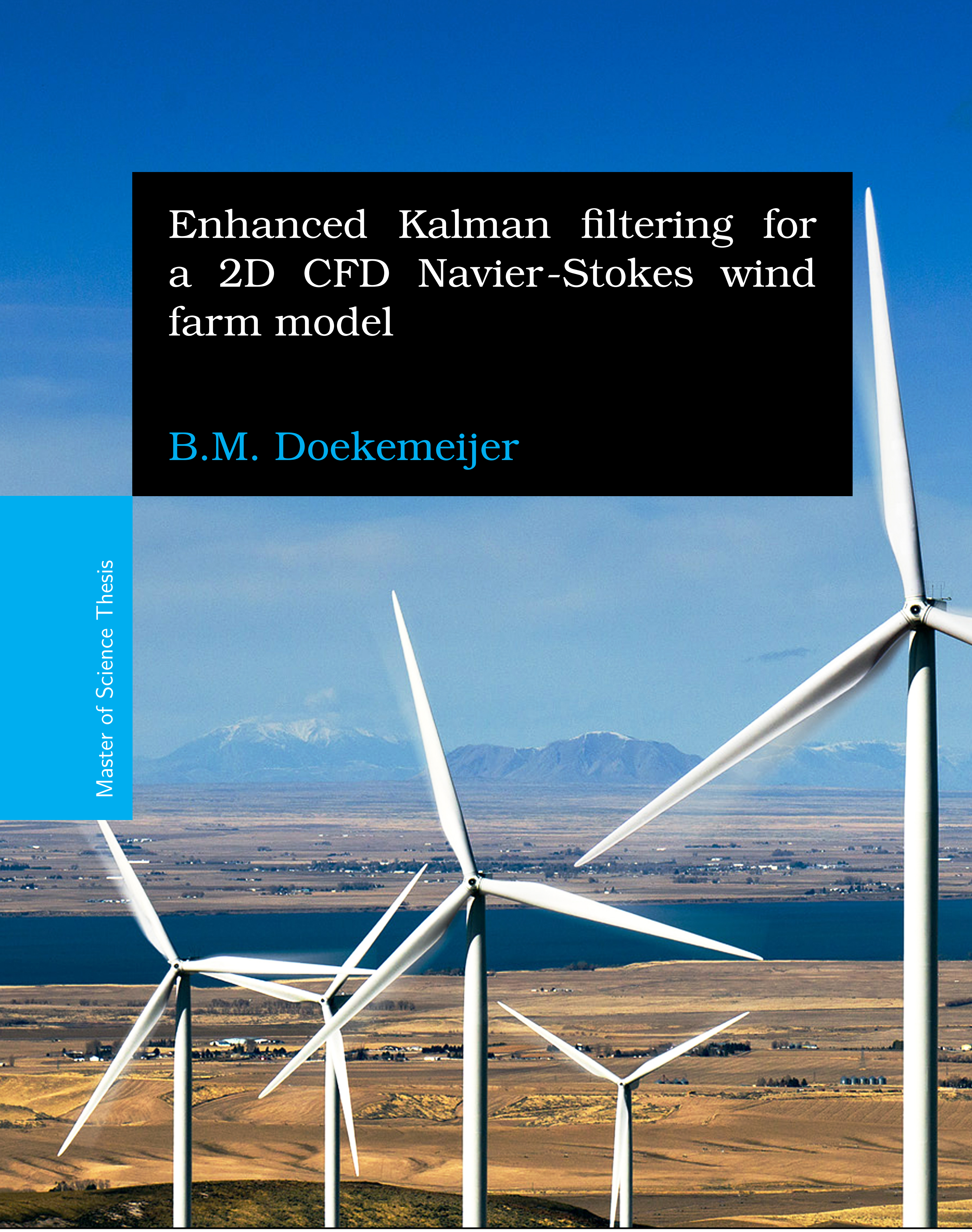
# Enhanced Kalman filtering for a 2D CFD Navier-Stokes wind farm model

B.M. Doekemeijer

Master of Science Thesis



Delft University of Technology

Delft Center for Systems and Control

# Enhanced Kalman filtering for a 2D CFD Navier-Stokes wind farm model

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

B.M. Doekemeijer

May 26, 2016

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis is supported by the University of Colorado Boulder (CU) and the National Wind Technology Center at the National Renewable Energy Laboratory (NREL). The frontpage photo is of Power County, Idaho, under the courtesy of the United States Department of Energy (DOE).

# Abstract

Wind turbines are often grouped together in wind farms for financial reasons, but due to wake development this results in decreased turbine lifetimes and power capture compared to an equal amount of individually placed turbines. Consequently, this results in an increased levelized cost of energy (LCOE). Wind farm control aims to minimize wind's LCOE by operating wind turbines at their optimal settings. Most state of the art control algorithms are open-loop and rely on a control-oriented, low fidelity, static flow model. Closed-loop control relying on a dynamic model has real potential to further decrease the LCOE of wind, but is often too computationally expensive for practical use.

In such a closed-loop framework, observers are essential because they improve estimations of the flow field based on a limited set of measurements. These algorithms correct for unmodeled dynamics (thus allowing the use of a lower fidelity, time-efficient flow model), noisy measurements and a limited number of measurements (spatially and temporally). Estimations of the current and future flow fields can be fed into control algorithms to calculate the optimal control policy for a wind farm.

In this thesis, two time-efficient state observers are designed using the control-oriented, medium fidelity, dynamic flow model WindFarmSimulator (WFSim), developed at the Delft University of Technology (TU Delft). WFSim is beneficial to current control-oriented models as it is applicable for different farm layouts and inflow conditions, it is dynamic, and it includes a higher fidelity flow model based on the Navier-Stokes (NS) equations. As real-time closed-loop control is the objective, the filter should take no more than 1 s per iteration.[1]

WFSim predicts the 2D velocity vectors in a horizontal plane at hub height in a wind farm over time. Fundamentally, it relies on a set of spatially and temporally discretized NS equations, which are reformulated as a nonlinear, implicit state-space system by projecting away the continuity equations. The first step in this work is to reformulate WFSim to allow direct implementation in state of the art filtering algorithms, resulting in a discrete-time, nonlinear, explicit state-space system. The farm model includes a mixing length turbulence model to account for wake recovery, and employs the actuator disk model to account for the rotor-flow

[1] Since WFSim will be simulated with a timestep of 1 s.

interaction. Boundary conditions are introduced by defining a certain inflow at one side of the grid, and enforcing zero flux conditions at the remaining three sides.

Based on WFSim, two time-efficient variants of the Kalman filter (KF) are implemented, namely an Approximate Kalman filter (ApKF) and an Ensemble Kalman filter (EnKF). The ApKF relies on the traditional KF algorithm, but improves time-efficiency by enforcing a sparsification of the main model matrices and state error covariance matrix. Simulations show negligible loss in performance, with gains in computation time by a factor $10^1 - 10^2$. In the EnKF, the covariance matrices are replaced by a sample covariance. As the sample covariance matrices are typically well described with several orders of magnitude less ensemble members, significant reductions in computational cost can be achieved. In this work, the EnKF has an even lower computational cost than the ApKF.

These state observers are then tested, including a case without observer (open-loop, i.e. model validation), using a dataset obtained from the high fidelity farm model Simulator fOr Wind Farm Applications (SOWFA), developed by the National Renewable Energy Laboratory (NREL) and TU Delft. Simulations are of a two turbine case in a 1000 by 2000 m domain size, meshed at two different resolutions: 25x50 grid points and 50x100 grid points, respectively. WFSim is simulated in open-loop (thus excluding a state observer) under settings that closely resemble the simulation setup of SOWFA. Good results are shown, with WFSim predicting an averaged flow in the wind farm. WFSim does neglect finer flow dynamics found in SOWFA due to its simplified turbulence model, simplified rotor model, and the absence of a vertical dimension (2D vs. 3D). However, in WFSim the wake width is overestimated, the wake depth is underestimated, wake recovery is underestimated, turbine hub effects are neglected, and wake meandering is not modeled. As expected, most estimation errors originate from the region near and behind the second, downstream turbine. State observers are implemented to account for these unmodeled dynamics.

In a similar setting, the two time-efficient filtering algorithms are assessed, with 23% and 11% of the system outputs available as measurements to the system[2] respectively for the 25x50 meshing and the 50x100 meshing. These measurements are flow velocities, disturbed by white noise with standard deviation $\sigma = 0.10$ m/s. Noticeably, the observers at the coarser meshing (25x50) perform very well, accounting for the overestimated wake width, the underestimated wake recovery, turbine hub effects, and wake meandering. Moreover, the EnKF even accounts somewhat for the underestimated wake depth. These filters only take 0.8 s and 0.7 s per iteration for the ApKF and the EnKF, respectively. For the EnKF, quantitatively, reductions in root mean square (RMS) error are up to 50% and improvements in the variance accounted for (VAF) are up to 24 percent points in the mean wake centerline.[3]

In conclusion, the ApKF and EnKF at a 25x50 meshing show very good performance at a fraction of the computational cost of standard filtering algorithms, accounting for unmodeled dynamics and noisy measurements. The EnKF is preferred over the ApKF due to its increased performance at a slightly lower computational cost. Furthermore, due to an iteration time of $\leq 1$ s, these algorithms already allow real-time implementation, and are expected to further improve when ported to machine code (e.g., `C`). The work presented in this thesis is a first major step to closed-loop control of wind farms.

[2] Measurements are spaced at equidistantly throughout the grid, and near the turbines, by Figure 4-1.

[3] The mean wake centerline is defined as the laterally averaged (from rotor end to rotor end) longitudinal flow velocity throughout the domain, by Figure 2-12.

# Table of Contents

# List of Figures

# List of Tables

# Preface

After consulting with my supervisor dr. ir. J.W. van Wingerden, the opportunity arose to perform my master's thesis abroad at the University of Colorado Boulder, located in the United States. Under local supervision of prof. L.Y. Pao in the department of Electrical, Computer and Energy Engineering, I performed research on wind farm modeling and control from September 2015 until May 2016. This report is the result of work performed from November 2015 onwards, in succession of my literature survey [1].

This work would not have been made possible without the support of the following people (in no particular order): Pieter Gebraad, currently employed by Siemens in Denmark and previously by the National Renewable Energy Laboratory (NREL) in the United States, Paul Fleming from NREL, Lucy Pao from the University of Colorado Boulder (CU), Jan-Willem van Wingerden from the Delft University of Technology (TU Delft), and Sjoerd Boersma from TU Delft. Their contributions in this work are greatly appreciated.

Delft University of Technology
May 26, 2016

B.M. Doekemeijer

# Chapter 1

# Introduction

This chapter introduces the topic at hand, its relevance and importance in current society, and more specifically what this research adds to the current body of literature. Firstly, the motivation for this work and, more generally, research in wind energy are outlined in Section 1-1. Secondly, background information on wind farm control and modeling will be summarized in Section 1-2, which will allow a more thorough understanding of the remainder of this thesis. Thirdly, in Section 1-3, the respective thesis objectives will be outlined more formally. Finally, the report structure will be depicted in Section 1-4.

## 1-1 Motivation

Ever since the industrial revolution in the early $19^{\text{th}}$ century, greenhouse gases and other human-induced pollution have severely impacted the environment. Global warming and climate change are perhaps the biggest human-inflicted threats to our species as we know it. With a seemingly ever-growing global demand for energy, the consequences will only increase in magnitude with time.[1] Consequences of global warming include extreme weather, resulting in e.g., large droughts [3] and thereby forest fires, risks for human health [4], melting of glaciers [5] and thereby the rise in sea level, and disturbances of ecological systems and biodiversity [6]. With a major part of The Netherlands being below sea level, the melting of glaciers will have a devastating impact on the country and its population. Hence, a global shift from climate change-inducing fossil fuels to green, renewable energy sources is of invaluable importance.

Fortunately, renewable energy alternatives are on the rise, with solutions in the fields of biomass, hydrogen, nuclear, solar and wind [7]. This report limits itself to the latter: energy from wind. Wind energy has shown serious potential in providing green energy (e.g., in Denmark, in which it provides for over 40% of the national energy demand [8]), but a number of challenges remain.

[1] The world energy demand is expected to grow by about one-third between 2013 and 2040, according to the recent annual report on energy of the International Energy Agency (IEA) [2].

For wind energy to be feasible, its levelized cost of energy (LCOE) has to be able to compete with that of other sources.[2] In the pursuit of minimizing wind's LCOE, wind turbines are typically grouped together in wind farms, saving on turbine and electricity grid deployment costs, operation and maintenance costs, land- or sea-usage and visual pollution, among others. However, placing turbines in each other's proximity gives rise to the development of, and interactions with, turbulent flow wakes, often resulting in a plant-wide decreased power capture[3] and increased turbine structural loading compared to an equal number of single turbines, thereby suppressing potential gains in terms of the LCOE of wind.

Figure 1-1 shows a photo of the Horns Rev offshore wind farm in Denmark operating in foggy conditions, taken by Christian Steiness in 2008. In this photo, the turbine wakes, their expansion downstream, and the overlap of wakes with downstream turbines can clearly be seen.

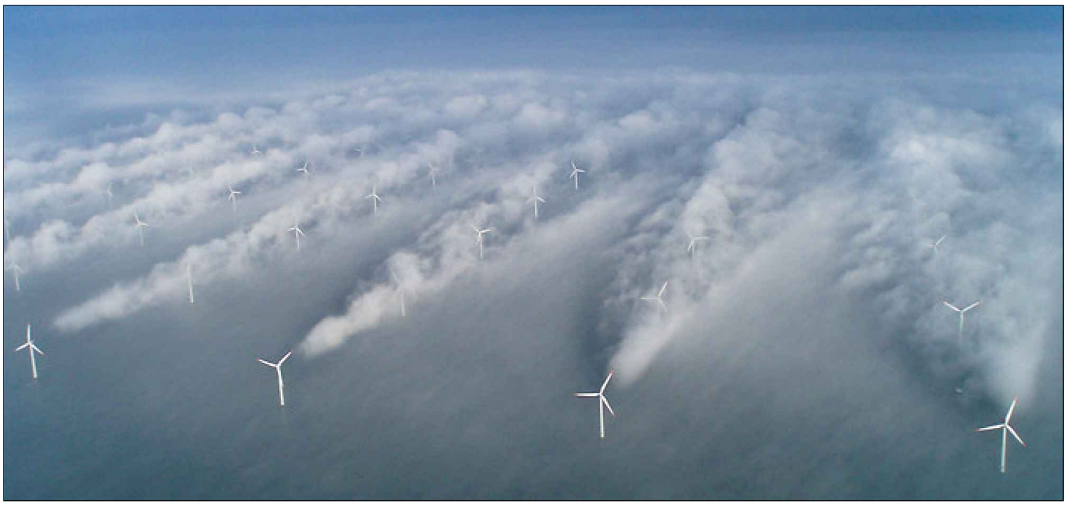


**Figure 1-1:** The Horns Rev offshore wind farm near Denmark under foggy weather conditions. The flow wakes formed by turbines extracting energy from the wind are clearly visible. [10]

Wind farm control aims to counter wake effects such as power capture losses and increased structural loading by operating turbines at their collectively optimal settings, thereby minimizing the LCOE of wind. Advancements in wind farm control have gone hand in hand with advancement in wind farm modeling, as state of the art control algorithms typically rely on a control-oriented, internal model of the flow. Unfortunately, these models are often of low fidelity, static, and are limited to open-loop applications. Furthermore, high fidelity dynamic simulation models based on the Navier-Stokes (NS) equations have been developed that allow relatively inexpensive controller testing and flow analysis, but are much too computationally costly for control applications. The nonlinear, time-varying, stochastic nature of the flow has significantly complicated flow modeling. Notably, the time-varying nature of the system requires fast solvers to make real-time control possible. A trade-off always has to be made between model fidelity and applicability in control.

[2] The levelized cost of energy (LCOE) is a quantitative measure to compare different methods of power generation in terms of financial feasibility. It incorporates the investment needed for realizing a specific energy generating solution, its operation, and total power output throughout its lifetime.

[3] According to some numerical studies [9], power capture losses in wind farms are predicted to be up to 50% for certain topologies and atmospheric conditions.

## 1-2 Basic concepts in wind farm modeling and control

For a good understanding of the work presented in this thesis, the reader is required to have at least a basic understanding of wind farm modeling and control. This section contains background information required for a clear comprehension of the work presented in Chapters 2 to 4, and a basic understanding of other state of the art research in the field. Note that an even more top-level introduction, the theory of wind and wind turbines, is briefly explained in Appendix A. Wind farm control and wind farm modeling are touched upon in this chapter. For a more thorough explanation including recent developments in these fields, please see the literature study preceding this work [1].

### 1-2-1 Wind farm control

Wind turbines are often grouped together in wind farms for a number of reasons [11, 12], including:

- Reduced deployment costs of the turbines and the electricity grid.
- Reduced operation and maintenance costs.
- Reduced land- or sea-usage and visual pollution.
- Increased power capture per unit area at resource-rich locations.

However, complications arise due to the development of wind flow wakes.[4] In wind farms, turbines typically operate in waked flow caused by their respective upstream turbines. As the flow in these wakes have decreased velocity and increased turbulence, downstream turbines capture less power and experience increased structural loading compared to when placed individually. Numerical studies indicate that power capture losses may be up to 50%, depending on the farm layout and atmospheric conditions [9]. The lifetime of downstream turbines is further decreased due to asymmetry in turbine loads [12].

Wind farm control typically aims to minimize power losses and structural loading by collectively controlling turbines in one or multiple wind farms.[5] Currently, wind turbines often follow a greedy-control approach, in which each turbine is operated to maximize their individual power capture.[6] While optimal for individual turbines, this has shown to be suboptimal in wind farms due to wake interactions. These control algorithms are typically enforced using the turbine degrees of freedom: generator torque $\tau_g$, turbine yaw $\gamma$, and rotor blade pitch angles $\beta$. For an elaborate, recent overview of the current state of the art in wind farm control, please see the literature survey by Knudsen *et al.* [14]. In this thesis, a concise overview is given of the two main control methodologies.

**Axial-induction-based control** In axial-induction-based control, the idea is to derate upstream turbines through blade pitch and generator torque, resulting in the development of less severe flow wakes. If the increase in power capture for downstream turbines exceeds the decrease in power capture for the upstream, derated turbines, axial-induction-based control is worthwhile.

[4] Please see Appendix A-1 for an introduction to wind flow wakes and their properties.

[5] Other objectives include e.g., providing stability to the electricity grid, as is the case in Ela *et al.* [13].

[6] This is for control region 2. See Appendix A-2 for an explanation of the different regions of control.

An important requirement for a successful implementation of axial-induction-based control is that the control settings adjust timely and accurately to changes in the inflow [14]. The amount of derating is traditionally calculated using a steady-state engineering model of the flow, and unnecessary derating due to erroneous predictions of inflow conditions may cause more losses than gains in axial-induction-based control.

Gebraad's Ph.D. dissertation [12] contains a literature review on axial-induction-based control. Based on this source and the survey by Knudsen *et al.* [14], it can be said that, using axial-induction-based control, theoretical improvements in power capture up to $4-6\%$ can be expected in idealized scenarios. However, in more practical settings, these works conclude that no definite statement can be made about its potential of increasing a farm's power capture.

**Wake redirection control** In wake redirection control, the rotors of upstream turbines are purposely misaligned with the incoming flow direction in order to deflect the wake downstream. This deflection can be done vertically (tilting) or horizontally (yawing). This control methodology has shown serious potential in high fidelity simulations with power increases of $4-7\%$ [14], and is still actively being researched. Fleming *et al.* [15, 16] suggested that a downwards deflection would promote wake recovery, but this degree of freedom is unfortunately not available on most modern wind turbines. With yaw being a common degree of freedom in modern wind turbines, literature has favored horizontal wake deflection. The concept is demonstrated in Figure 1-2, in which $U_\infty$ depicts the freestream (inflow) wind speed.

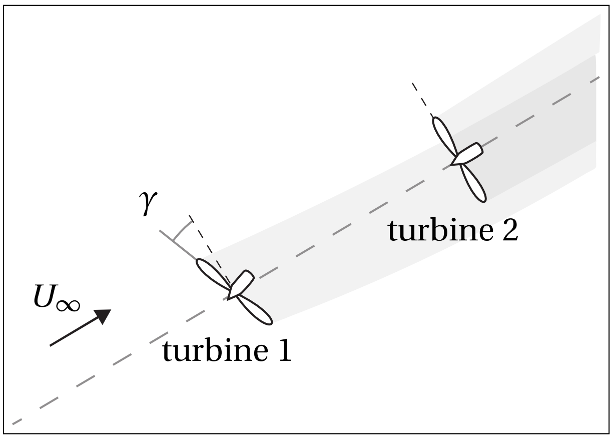


**Figure 1-2:** Wake redirection control as demonstrated in Gebraad's dissertation [12], with wakes in gray. Upstream turbines are purposely misaligned to deflect their wakes downstream, resulting in no or partial overlap with downstream turbines.

Due to rotor misalignment of the first turbine with the inflow, the wake is redirected downstream and thereby no longer, or only partially, overlaps with a downstream turbine, resulting in increased power capture. Partial overlap of a flow wake with downstream turbines does increase structural loading, but this can be mitigated by e.g., individual pitch control (IPC). IPC has shown to be able to reduce structural loads significantly, e.g., in high fidelity simulations by Wilson *et al.* [17], showing blade root flap bending moment reductions in standard deviation from the mean of 14 – 32%. More information on IPC can be found in work by Bossanyi (e.g., [18, 19]), and in Knudsen *et al.* [14].

## 1-2-2 Wind farm modeling

Advancements in wind farm control have gone hand in hand with advancements in wind farm modeling, as state of the art control algorithms typically rely on an internal model of the flow. Common properties of such a model are that it is control-oriented, static, of low to medium fidelity, and often relies on heuristic data rather than a body of equations derived from physics.

Next to the development of low fidelity control-oriented models, high fidelity flow models have allowed wind farm simulations at high accuracy. At a fraction of the cost of field tests, these models have allowed flow analysis and controller testing (e.g., in work by Fleming *et al.* [16, 20]), among others. These models are significantly more computationally expensive than their control-oriented counterparts, and therefore are not applicable for real-time control.

Many wind farm models exist, varying in accuracy, applicability and computational cost. Sanderse *et al.* [21] provides an overview of rotor blade models and wake models with a focus on computational fluid dynamics (CFD) methods. Next, rather a concise overview is given of the general framework of a wind farm model.

### General model framework

Generally, wind farm models consist of three main components, namely:

- Rotor model – This submodel predicts the turbine-flow interactions. This includes both forces by the flow on the turbine, and forces by the turbine on the flow. A common rotor model is the actuator disk model (ADM), inspired by momentum theory [22].
- Turbine model – This submodel predicts the dynamics and structural loading on a turbine under an incoming flow field. Typically included in a turbine model are extreme load cases, vibrational modes and fatigue loads. A popular high fidelity turbine submodel is the Fatigue, Aerodynamics, Structures and Turbulence (FAST) model, developed at the National Renewable Energy Laboratory (NREL) [23].
- Flow model – This submodel predicts the flow properties (e.g., velocity, direction, turbulence) in a wake or in a flow field, either dynamically or as a steady-state solution. Popular examples of low fidelity flow models are Jensen [24] and Ainslie [25], commonly used for control purposes. A popular high fidelity simulation model is Simulator fOr Wind Farm Applications (SOWFA), developed at NREL [26]. A simulation of several minutes in SOWFA can take multiple days on NREL's supercomputer "Peregrine". It is used for low fidelity model validation, high fidelity controller testing and flow analyses.

An important comment to make is that not all components are required to be present in a farm model. Rather, different components can be simulated individually to assess one certain aspect of the wind farm, while others are omitted. For example, for a predefined inflow case, one may only look at the turbine model to assess structural degradation under cyclic and extreme loading, without assessing the flow propagation and wake development downstream.

### Overview of wind farm models

In the literature review preceding this thesis [1], a more detailed analysis was performed on wind farm modeling. An overview was made of different wind farm models, their characteristics and applications. The result is summarized in Table 1-1.

**Table 1-1:** Wind farm models overview.

| | **Low fidelity** | | **Medium fidelity** | **High fidelity** |
|---|---|---|---|---|
| Model type | Kinematic models | Flow field models | Flow field models | Flow field models |
| Fundamentals | Heuristic | RANS | 2D (un)steady NS | 3D unsteady NS |
| Models | Jensen, FLORIS, Frandsen, ... | Ainslie, ... | FLORIDyn, DWM model, WFSim, ... | SOWFA, WakeFarm, ... |
| Flow dimension | 2D | | 2D/3D | 3D |
| Dynamic/Static | Static | | Dynamic | Dynamic |
| Rotor model | ADM | | ADM | ADM/ALM |
| Turbine model | Static relationships from actuator disk theory | | Static relationships or an aero-elastic package (e.g., FAST) | |
| Comp. effort | Order of seconds on a desktop PC | | Order of minutes on a desktop PC | Order of days on a cluster of $10^2$ CPUs |
| Model accuracy | Low – medium | | Medium – high | High – very high |

The models are sub-divided under kinematic models and field models. Kinematic models rely on the momentum equation to describe the waked flow, and thereby disregard the near wake region and changes in turbulence. Traditionally, these models require to be coupled with a turbulence model for this purpose. On the other hand, field models predict the entire flow field based on the Navier-Stokes equations, often also coupled with turbulence models.

In this table, Reynolds Averaging of Navier-Stokes (RANS) models neglect turbulent dynamics in the flow and predict an averaged flow velocity and direction, while full NS models attempt to include turbulence. Low fidelity models include FLOw Redirection and Induction in Steady-state (FLORIS) from the Delft University of Technology (TU Delft), and its more recent dynamic extension FLOw Redirection and Induction Dynamics (FLORIDyn), which is considered of medium fidelity. Another popular medium fidelity flow model is the Dynamic Wake Meandering (DWM) model from the Technical University of Denmark (DTU). SOWFA is a high fidelity simulation model from TU Delft and NREL, and will be discussed more in detail in Section 2-2. SOWFA includes the actuator line model (ALM) for rotor modeling, which is a more sophisticated variant of the aforementioned ADM.

What this table shows is that choosing the right model for one's purpose is not a trivial task. Trade-offs are to be made on many fronts. Roughly two distinct goals have been and are being pursued in literature: highest accuracy at practically any computational cost for simulation, and highest accuracy at a limited computational cost applicable for control.

## 1-3 Thesis objectives

The work in this thesis is related to the medium fidelity flow model WindFarmSimulator (WFSim) from the Delft University of Technology, first introduced in a paper by Torres *et al.* in 2011 [27], and later improved in a paper by Boersma *et al.* [28]. This model is a dynamic 2D flow model based on a set of spatially and temporally discretized NS equations. Models such as WFSim aim to bridge the gap between low fidelity, control-oriented models and high fidelity simulation models. WFSim is an attempt to combine the ease of application in control as found in low fidelity models, with high accuracy as found in high fidelity models. Mainly, WFSim is superior to low fidelity models because it is dynamic, lending itself to closed-loop control. Furthermore, it is applicable for different farm layouts, wind directions, wind speeds and atmospheric conditions. The concept of closed-loop control using WFSim as pursued by the current research group is demonstrated in Figure 1-3.

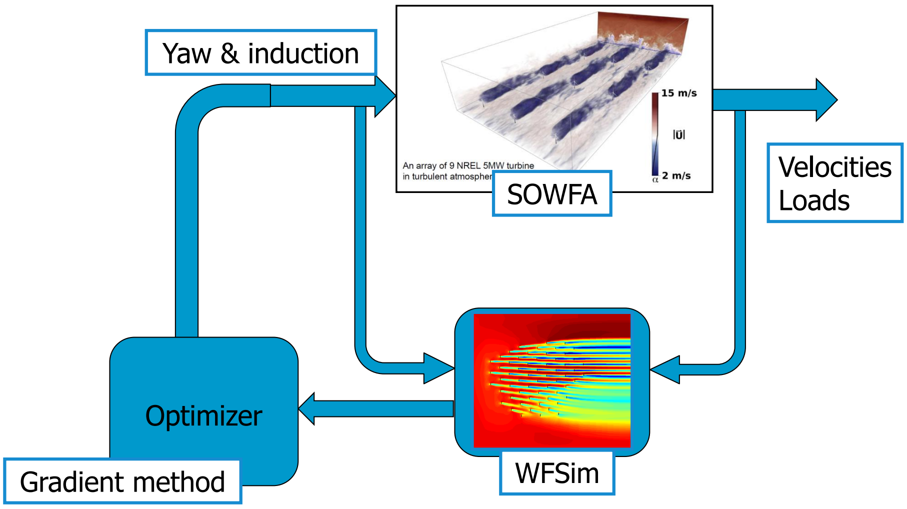


**Figure 1-3:** Closed-loop control for wind farms. The main focus in this thesis lies on observer design using the medium fidelity WFSim flow model. Image by J.W. van Wingerden.

In this figure, the flow of information is displayed. Physical measurements in a wind farm (e.g., velocities, turbine loads) are fed into the observer, which internally relies on the 2D flow model WFSim. The observer filters the measurements and provides an estimate of measured and non-measured flow properties (e.g., velocities) at the current time instant. Furthermore, the observer can be used to predict flow properties ahead in time. Then, this information is fed into an “optimizer” algorithm, which calculates the optimal control policy to achieve a certain performance goal (e.g., maximum power capture, minimal turbine loading). This control policy is then enforced on the wind farm in real-time. Finally, time propagates, new measurements are obtained, and the cycle repeats itself. Note that in the current scheme, the true wind farm is replaced by data from the high fidelity simulation model SOWFA.[7]

The benefit of this scheme with respect to open-loop control lies in the nonlinear, time-varying nature of air flows behind wind turbines. Accurate predictions would require an increasingly sophisticated flow model, and thereby an increasing number of computational resources. Additionally, these models will still not capture all dynamics as found in a true wind farm due to the complex dynamics found in turbine-flow interactions and wind wakes. An observer has to potential to correct for unmodeled flow dynamics and modeling error, while relying on a simplified, time-efficient flow model.

Several parties are currently working with WFSim. Boersma and Van Wingerden[8] are actively developing the WFSim model, with support of Vali[8] and Doekemeijer[7]. Recent extensions to the model include yaw actuation, a mixing length turbulence model and a linearized version of WFSim [28, 29]. Furthermore, Kühn and Vali[9] are in the process of developing control algorithms for optimal energy extraction using an adjoint-based gradient of a cost function derived from WFSim [30].

The main objective of the work presented in this thesis is to design a robust state observer with low computational cost, enabling real-time closed-loop control of wind farms. Observers as such are important for a number of reasons:

- There is always an error between model and reality which needs to be corrected for. With the complex dynamics in wind wakes, a state observer is of invaluable importance for proper flow field reconstruction. A model cannot capture all these dynamics, especially with limited computational resources, and furthermore often many model parameters are uncertain.
- Measurements always include some noise, and are desired to be filtered.
- The flow field is only measured at a few instances in the wind farm, while the entire flow field may be of interest. It is impractical and infeasible to measure the velocity at each point in the grid. A state observer will predict the flow velocities at the remaining grid points based on the internal model and available measurements.

Quantifying computational efficiency: the objective is to have a computational time of no more than a certain percentage of the sampling time of the simulation. In this thesis, since WFSim will be sampled at 1 Hz, the summed computation time of the observer and control algorithm should be no more than 1 s in order to allow real-time control of the farm. Thus, in a limit case, it is desired to have the observer provide state estimations at a maximum iteration time of 1 s.

Secondary objectives essential to this work include the validation of the WFSim model with high fidelity simulation data and optimization of WFSim's numerical implementation.

[7] Note that the work in this thesis is limited to axial-induction-based control, and conclusions do not necessarily hold for wake redirection control. Recent versions of WFSim do include the yaw degree of freedom to allow wake redirection control. While no obstructions are expected when moving from axial-induction-based control to wake redirection control using the observers presented in this thesis, it has not yet been assessed through experiments due to limited availability of high fidelity data. Future work may include the implementation and assessment of wake redirection control using WFSim.

[8] Delft Center for Systems and Control (DCSC), Delft University of Technology, The Netherlands.

[9] Wind Energy System (WESys) Research Group, ForWind, University of Oldenburg, Germany.

## 1-4 Structure of the report

With the motivation, fundamental concepts and thesis objective introduced in this chapter, Chapter 2 goes further into detail on the WFSim model. This chapter also includes model validation of WFSim using a high fidelity simulation model (SOWFA). Then, in Chapter 3, two enhanced Kalman filter designs are outlined, each following a fundamentally different approach to reduce computational effort and improve filtering results. Simulation results for both state observers are outlined in Chapter 4, after which conclusions are drawn in Chapter 5. Recommendations and a future outlook for this research will also be touched upon in this chapter.

The appendices include an introduction to wind and single turbine control in Appendix A, simulation results for the lateral flow components in Appendix B, and a paper submission of the work presented in this thesis in Appendix C. The report is concluded with a list of references, list of acronyms and list of symbols.

Chapter 2

# WFSim and model validation using SOWFA

This chapter elaborates on the WindFarmSimulator (WFSim) model developed at the Delft University of Technology (TU Delft) [27, 28]. This model lays the fundamentals for the state observers, which will be described in Chapter 3. WFSim is the subject of Section 2-1. Secondly, the higher fidelity farm model SOWFA will be described concisely in Section 2-2. Thirdly, SOWFA is used for model validation of WFSim in Section 2-3.

## 2-1 The WFSim flow model

In this section, the WFSim model will be described. First, an overview will be given for the reader with time restrictions in Section 2-1-1. In the remainder of the section, the model will be described in detail. Model assumptions are detailed in Section 2-1-2, after which the grid layout is described in Section 2-1-3. The Navier-Stokes (NS) flow equations will be depicted in Section 2-1-4, after which the turbulence submodel will be described in Section 2-1-5. The rotor submodel will be the topic of Section 2-1-6. The final form of the model and the boundary conditions are obtained in Sections 2-1-7 to 2-1-9. Software implementation and benefits & challenges in the model's application will be depicted in Sections 2-1-10 and 2-1-11, respectively.

### 2-1-1 A brief overview

In short, WFSim predicts the flow velocity vectors throughout a wind farm in a predefined meshing using the 2D Navier-Stokes (NS) flow equations in a computational fluid dynamics (CFD) formulation. It is considered a medium fidelity dynamic flow model. It employs the actuator disk model (ADM) to calculate the aerodynamic forces exerted on the flow by the rotor blades. Furthermore, it includes a mixing length wake turbulence model to account for wake recovery. The velocity vectors are predicted in a horizontal plane at turbine hub height.

The spatially and temporally discretized NS equations are formulated as a state-space system by projecting away the continuity equations. Following this process, a discrete-time quasi-linear parameter varying (qLPV) state-space system formulation in implicit form is obtained, described by

$$\begin{aligned} E_w(\alpha_k)\alpha_{k+1} &= A_w\ \alpha_k + B_w(\alpha_k)\beta_{w_k} + S_w(\alpha_k), \\ \begin{bmatrix} \overline{u}_k \\ \overline{v}_k \end{bmatrix} &= Q_p\ \alpha_k + B_p. \end{aligned}$$

In this equation, $\alpha_k \in \mathbb{R}^N$ represents the system state vector at time $k$, resulting from the projection of the 2D velocity vectors $[\overline{u}_k; \overline{v}_k] \in \mathbb{R}^M$. Note that the entries of $\alpha_k$ are not physical velocities due to the projection. $N$ is proportional to the number of grid points with a typical value of $10^3$, giving a measure of refinement in flow field estimates. $0 \leq \beta_{w_k} \leq 1 \in \mathbb{R}^O$ is the system input; a vector with $O$ rows, where $O$ is equal to the number of turbines. $\beta_{w_k}$ is a scaled measure of the axial induction factors $a_k$, representing how much energy is extracted from the wind by each turbine, by $\beta_{w_k} = a_k\,(1 - a_k)^{-1}$. From $\beta_w$, the longitudinal and lateral aerodynamic forces that the turbines exert on the flow are calculated. $\beta_w$ can be translated into physical turbine settings such as generator torque, blade pitch and yaw. Furthermore, the system matrices are represented by $A_w \in \mathbb{R}^{N\times N}$, $B_p \in \mathbb{R}^M$, $B_w \in \mathbb{R}^{N\times O}$, $E_w \in \mathbb{R}^{N\times N}$, $Q_p \in \mathbb{R}^{M\times N}$ and $S_w \in \mathbb{R}^N$, of which $E_w$, $B_w$ and $S_w$ depend on $\alpha_k$. These system matrices are sparse and structured, and can thus be exploited for computational efficiency.

Boundary conditions for both longitudinal and lateral velocity are introduced by defining a certain inflow at one side of the grid, and enforcing zero flux conditions at the remaining three sides.

### 2-1-2 Assumptions

In the remainder of this section, the model is described on a level that is sufficient to obtain a good understanding of the capabilities and limitations of WFSim. A full derivation of the model requires a good understanding of CFD methods, and is therefore excluded from this report. However, an attempt is made to give an elaborate description without going into detail on computational fluid dynamics.

A number of assumptions are made in the WFSim model. Firstly, the flow is assumed to be incompressible, and thus with constant density [28]. Secondly, the vertical velocity component is neglected, and only a 2D set of NS equations at hub height is considered. Momentum theory is used to predict the turbine-flow interactions, and a simple mixing length wake turbulence model is implemented to account for wake recovery.

WFSim relies on a spatial and temporal discretization of the simulation domain, thereby resulting in a large number of cells for which a set of discrete NS equations are to be solved at each time instant. First, the way in which the domain is discretized spatially is demonstrated, after which the relevant equations will be derived and discussed.

### 2-1-3 Grid layout

Two types of grids are implemented in WFSim: one with linear (equidistant) spacing (in which all cells are of equal size), and one with exponential spacing (in cells are of different size). The latter is computationally beneficial, as higher accuracy can be achieved at areas of interest (i.e., around the turbines and wakes) with similar or lower computational costs than in an equidistantly spaced grid. However, for simplicity and demonstration purposes, a linear grid is applied in this section.

For any simulation, two subgrids in WFSim are implemented upon which the pressure terms, longitudinal velocity components, and lateral velocity components are expressed. The first subgrid is spaced uniformly covering the entire mesh, with longitudinal coordinates $X_1 \in \mathbb{R}^{N_x}$ and lateral coordinates $Y_1 \in \mathbb{R}^{N_y}$, according to

$$X_1 = \begin{bmatrix} 0 & 1 \cdot \frac{L_x}{N_x - 1} & 2 \cdot \frac{L_x}{N_x - 1} & 3 \cdot \frac{L_x}{N_x - 1} & \cdots & L_x \end{bmatrix}, \tag{2-1}$$

$$Y_1 = \begin{bmatrix} 0 & 1 \cdot \frac{L_y}{N_y - 1} & 2 \cdot \frac{L_y}{N_y - 1} & 3 \cdot \frac{L_y}{N_y - 1} & \cdots & L_y \end{bmatrix}, \tag{2-2}$$

with $L_x$ and $L_y$ the longitudinal and lateral lengths of the rectangular domain that is to be discretized, and $N_x$ and $N_y$ the total number of mesh points in longitudinal and lateral direction, respectively. The second subgrid with $X_2 \in \mathbb{R}^{N_x}$ and $Y_2 \in \mathbb{R}^{N_y}$ is spaced such that

$$X_2 = X_1 + \frac{1}{2} \cdot \frac{L_x}{N_x - 1}, \tag{2-3}$$

$$Y_2 = Y_1 + \frac{1}{2} \cdot \frac{L_y}{N_y - 1}, \tag{2-4}$$

resulting in each grid point of the second subgrid being centered within its four surrounding grid points of the first subgrid. Now, the pressure terms $p$ are gridded on $(X_1, Y_1)$, the longitudinal velocities $u$ are gridded on $(X_2, Y_1)$, and the lateral velocities $v$ are gridded on $(X_1, Y_2)$. If this would not be the case, there would be problems in the solution for pressure terms [31].[1] An example of a linear grid is displayed in Figure 2-1, with $L_x = 60$ m, $L_y = 40$ m, $N_x = 4$, and $N_y = 3$.[2]

For each cell, with $(I, J)$ corresponding to $(X_1, Y_1)$ and $(i, j)$ corresponding to $(X_2, Y_2)$, $u(i, J)$ is calculated with dependency on $u(i-1, J)$, $u(i+1, J)$, $u(i, J-1)$, $u(i, J+1)$, $p(I, J)$, $p(I+1, J)$, $v(I, j)$, $v(I, j-1)$, $v(I+1, j)$, and $v(I+1, j-1)$, which basically comes down to the 4 neighboring grid points for $u$ and $v$, and the 2 neighboring grid points of $p$ in longitudinal direction. Similarly, for $v(I, j)$ the dependency is on $v(I-1, j)$, $v(I+1, j)$, $v(I, j-1)$, $v(I, j+1)$, $p(I, J)$, $p(I, J+1)$, $u(i, J)$, $u(i, J+1)$, $u(i-1, J)$, and $u(i-1, J+1)$. Finally, for $p(I, J)$, the dependency is on $u(i, J)$, $u(i-1, J)$, $v(I, j)$, and $v(I, j-1)$.

Boundary conditions are introduced on the outer rows and columns of the grid, to ensure the existence of a solution for the NS equations. First, the NS equations will be described next. The boundary conditions will be explained in Section 2-1-9.

[1] If the same grid would be used for pressure terms and velocity, information could be lost resulting in an erroneous discretization.

[2] Two examples of an exponential grid can be seen in Figure 4-1. These grids will be used for model validation and observer testing.

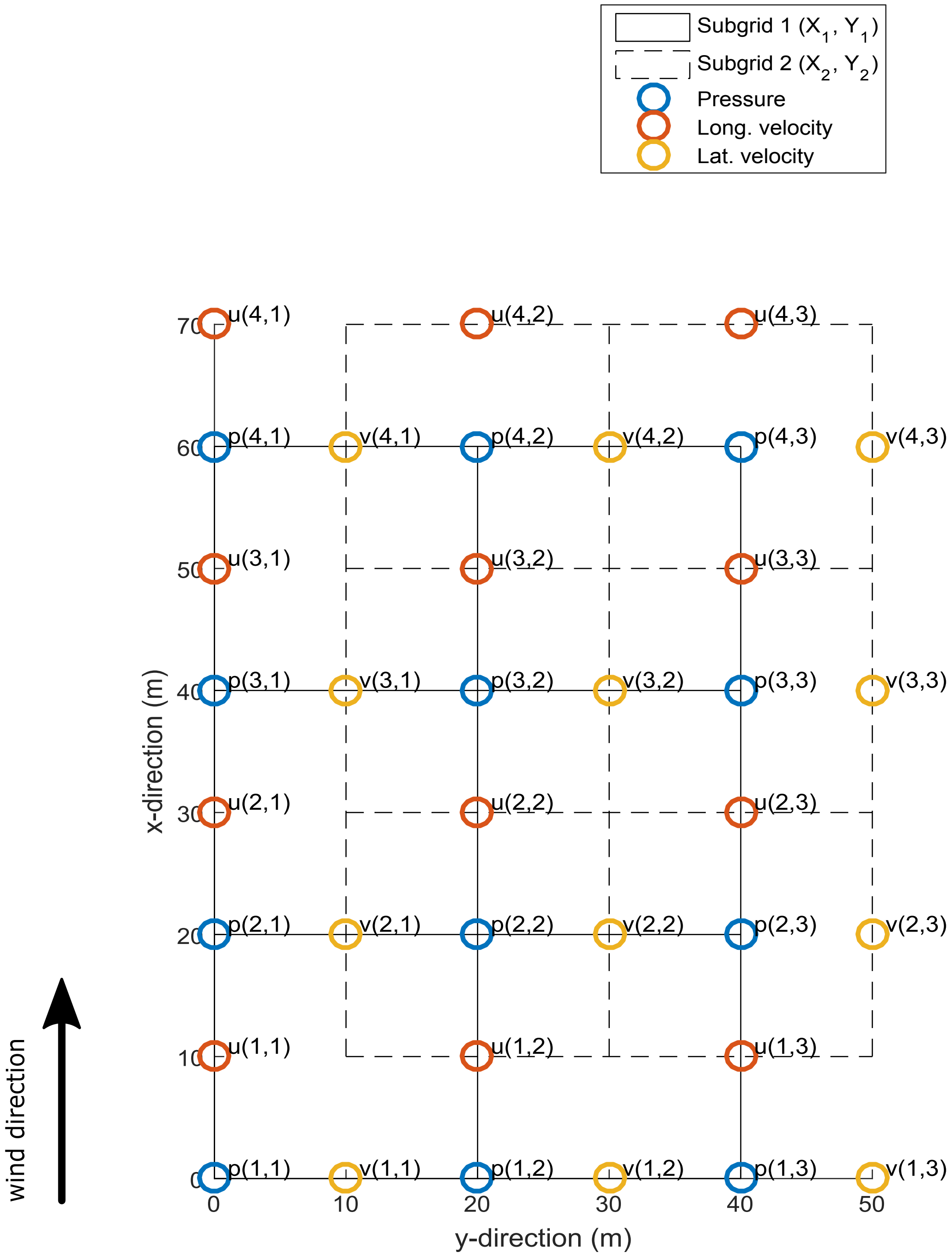


**Figure 2-1:** Example of a linear grid layout for the WFSim farm model, with $L_x = 60$ m, $L_y = 40$ m, $N_x = 4$, $N_y = 3$. The primary grid is represented in solid black, and the secondary grid is represented using dashed lines. The wind is flowing from bottom to top.

### 2-1-4 Navier-Stokes equations

For each cell in the grid, there are 3 equations to be solved originating from the set of Navier-Stokes (NS) equations assuming pressure terms to be constant, namely

$$\underbrace{\rho\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)}_{\text{Rate of change of total mass in the system}} = 0, \tag{2-5}$$

$$\rho\left(\underbrace{\frac{\partial u}{\partial t}}_{\text{variation}}+\underbrace{\frac{\partial u^2}{\partial x}+\frac{\partial u\cdot v}{\partial y}}_{\text{convection}}\right) = -\underbrace{\frac{\partial p}{\partial x}}_{\text{pres. grad.}} +\mu\left(\underbrace{\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}}_{\text{diffusion}}\right) + \underbrace{S_{M_u}}_{\text{ext. force}} + \underbrace{T_{M_u}}_{\text{turbulence}}, \tag{2-6}$$

$$\rho\left(\underbrace{\frac{\partial v}{\partial t}}_{\text{variation}}+\underbrace{\frac{\partial v^2}{\partial y}+\frac{\partial v\cdot u}{\partial x}}_{\text{convection}}\right) = -\underbrace{\frac{\partial p}{\partial y}}_{\text{pres. grad.}} +\mu\left(\underbrace{\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}}_{\text{diffusion}}\right) + \underbrace{S_{M_v}}_{\text{ext. force}} + \underbrace{T_{M_v}}_{\text{turbulence}}, \tag{2-7}$$

which entail conservation of mass, conservation of momentum in longitudinal direction, and conservation of momentum in lateral direction, respectively. In these equations $u$ and $v$ entail the velocities in longitudinal and lateral direction in m/s, $\rho$ is the flow density in N·s$^2$·m$^{-4}$, $\mu$ is the flow viscosity in N·s·m$^{-2}$, $S_{M_u}$ and $S_{M_v}$ represent turbine forces on the flow in longitudinal and lateral direction in N·m$^{-3}$, $T_{M_u}$ and $T_{M_v}$ are the longitudinal and lateral turbulence terms in N·m$^{-3}$, and $p$ is a pressure term in N·m$^{-2}$. This problem is a nonlinear partial differential equation (PDE) without analytical solution, thus requiring discretization. First, the turbulence and rotor model are described in Sections 2-1-5 and 2-1-6, after which the discretization will be described in Section 2-1-7.

### 2-1-5 Turbulence model

A simple turbulence submodel is included in WFSim to account for wake recovery: the mixing length wake turbulence model [32]. Implementation was outlined previously by Gebraad in 2012, and is repeated here [33].

Turbulence is only modeled in longitudinal direction, as in this way it does not alter the structure of the NS equations, and the same solution methodology can be followed. Thus, $T_{M_v} = 0$. This means wake recovery is lateral direction is neglected. However, dominant dynamics and the main interest for control are in longitudinal flow, and thus this assumption is deemed acceptable for its current purposes.

In the case that turbulence would be modeled also in lateral direction, the turbulence terms in Equations (2-6) and (2-7) would be described by

$$T_{M_u} = \frac{\partial \tau_{uu}}{\partial x}+\frac{\partial \tau_{uv}}{\partial y}, \tag{2-8}$$

$$T_{M_v} = \frac{\partial \tau_{vu}}{\partial x}+\frac{\partial \tau_{vv}}{\partial y}, \tag{2-9}$$

with Reynolds normal stresses $\tau_{uu}$ and $\tau_{vv}$, and Reynolds shear stresses $\tau_{uv}$ and $\tau_{vu}$. The Reynolds stresses are calculated according to

$$\begin{aligned} \tau_{uu} &= -\rho\overline{u'^2}, && \text{(2-10)} \\ \tau_{vv} &= -\rho\overline{v'^2}, && \text{(2-11)} \\ \tau_{uv} &= \tau_{vu} = -\rho\overline{u'v'} && \text{(2-12)} \end{aligned}$$

with $u'$ and $v'$ the longitudinal and lateral velocity fluctuations from turbulence. The overbar denotes the time-averaging operation

$$\overline{\phi} = \frac{1}{\Delta t}\int_0^{\Delta t} \phi(t)dt. \tag{2-13}$$

Following the Boussinesq assumptions in combination with the mixing length model, the approximations made for the Reynolds stresses are

$$\begin{aligned} \tau_{uu} &= \tau_{vv} \approx 0, && \text{(2-14)} \\ \tau_{uv} &= \tau_{vu} \approx \rho l_m^2 \left|\frac{\partial u}{\partial y}\right| \frac{\partial u}{\partial y}, && \text{(2-15)} \end{aligned}$$

with $l_m^2$ the eddy length scale. This variable is considered a tuning parameter, and is typically approximated by $l_m \approx 0.16 \cdot L_w$, with $L_w$ the wake half width.

Equation (2-8) is substituted in Equation (2-6) and will follow the same spatial and temporal discretization procedure as the other terms, to be depicted in Section 2-1-7. The turbulence term $T_{M_v} = 0$, and is currently neglected from Equation (2-7), thereby neglecting wake recovery in lateral direction. Including the expression for $T_{M_v}$ in Equation (2-7) would introduce extra coupling between $u$ and $v$ in this equation. It is currently still ambiguous how to incorporate these additional terms in the current solution methodology. However, the assumption is deemed acceptable due to the small contribution of lateral flow to the quantities of interest: the longitudinal velocities in the wake.

This model is a first step to turbulence modeling, and introduces wake recovery in WFSim. Currently, it is only accurate for slow changes of the flow direction, as transport of turbulence due to convection and diffusion are not taken into account in the mixing length model.

### 2-1-6 Rotor model

The rotor model employed by WFSim is the actuator disk model (ADM), which is based on momentum theory. A more elaborate discussion on this model is presented next, largely inspired by the book of Bianchi *et al.* [22]. Note that this work is solely focused on axial-induction-based control, and thus changes due to turbine yaw are not incorporated into the following equations.

In the actuator disk model, the turbine rotor plane is approximated by an infinitely thin homogeneous actuator disk extracting energy uniformly from the wind. It estimates the captured power by the turbine in a relatively simple way, and provides a theoretical upper bound for the amount of energy that can be extracted by a wind turbine. Consider Figure 2-2,

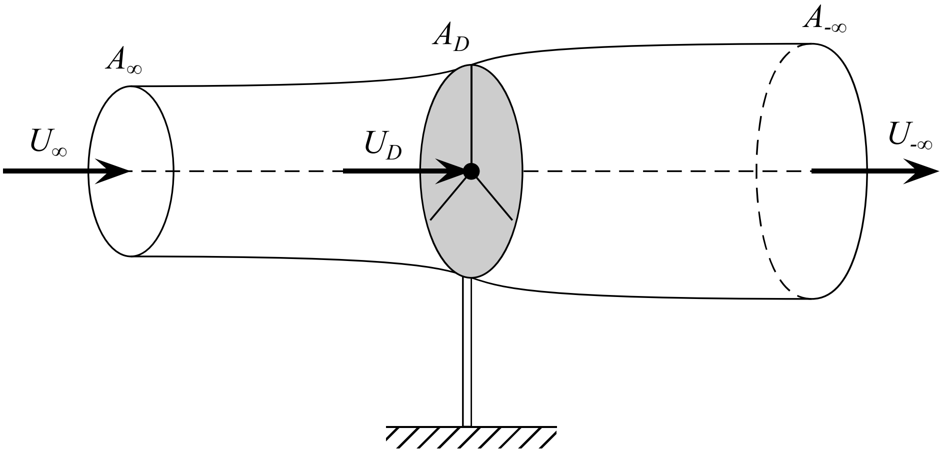


**Figure 2-2:** Actuator disk model: flow across an infinitely thin disk [22].

which demonstrates the wind flow through the infinitely thin disk according to momentum theory.

By the conservation of mass and under the assumption of incompressible flows, the flow rates at each cross-section are equal: $U_{\infty}A_{\infty} = U_D A_D = U_{-\infty}A_{-\infty}$, with $U_{\infty}$, $U_D$ and $U_{-\infty}$ the velocities far upstream of, at, and far downstream of the disk, respectively.[3] $A_{\infty}$, $A_D$ and $A_{-\infty}$ are defined similarly to be the cross-sectional areas at the corresponding locations. Caused by a difference in pressure before and after the actuator disk, the force on the flow $F_D$ at this location is calculated to be

$$F_D = (U_{\infty} - U_{-\infty})\,\rho A_D U_D, \tag{2-16}$$

and the flow velocity at (or rather: just downstream of) the disk $U_D$ is calculated by

$$U_D = (1-a)\,U_{\infty}. \tag{2-17}$$

In Equation (2-17), $a$ is the axial induction factor, indicating how much energy is extracted by the disk. As energy is extracted by the rotor, the wind speed further decreases and turbulent forces are introduced in the flow. Applying Bernoulli's equation on the upstream and downstream flow, the wind velocity downstream is

$$U_{-\infty} = (1-2a)U_{\infty}. \tag{2-18}$$

Equation (2-18) provides a theoretical lower limit for the wind velocity downstream of a turbine. Furthermore, the cross-sectional area of a waked region typically increases with downstream distance due to conservation of mass and a steady decay in flow speed.[4]

In WFSim the forces introduced by the turbine on the flow are modeled by a single point thrust force at each grid point where the turbine is defined. Using Equations (2-16) to (2-18), $F_D$ is expressed as

$$F_D = 2\rho A_D U_{\infty}^2 a(1-a) \tag{2-19}$$

[3] In real wind farms, the location and value of $U_{\infty}$ and $U_{-\infty}$ are not well defined.

[4] An example of a predicted flow including expanding wake using the ADM in WFSim can be found in Appendix A, Figure A-1.

with $A_D$ in practice being the rotor swept area. The ratio between power in the wind $P_V$ and captured power $P$, the power coefficient $C_P$, is defined by momentum theory as

$$C_P = \frac{P}{P_V} = 4a\,(1-a)^2 \tag{2-20}$$

with $a$ the axial induction factor of the relevant turbine. In WFSim, the input to the quasi-linear parameter varying (qLPV) state-space system obtained after discretization (to be discussed in Sections 2-1-7 and 2-1-8) will be $\beta_{w_k}$, calculated by

$$\beta_{w_k} = \frac{a_k}{1-a_k}, \tag{2-21}$$

which can be translated into practical wind turbine settings such as turbine yaw, generator torque and blade pitch angles.

An important remark is that in reality, energy is extracted through turbines by a torque on the rotor blades. For more information about these derivations, see the book by Bianchi *et al.* [22]. Also note that these equations hold for non-yawed conditions. However, extensions do exist in literature to account for rotor yaw, e.g., in Gebraad's Ph.D. thesis [12], and are also implemented in recent versions of WFSim.

**Betz limit** Betz limit is the theoretical maximum power capture ratio $C_{P_{\max}}$, obtained by taking the derivative of Equation (2-20) with respect to $a$. The Betz limit states that the highest axial induction factor achievable is $a = 1/3$, translating into $C_{P_{\max}} = 16/27 \approx 0.59$. For WFSim, this would translate into an input signal $\beta_w = 0.50$. In practice, the maximum power coefficient for horizontal-axis wind turbines typically lies around 0.45 [22].

### 2-1-7 Spatial and temporal discretization

Using the finite element method, Equations (2-5) to (2-7) are discretized spatially using the Hybrid differencing scheme, resulting in a discretized form of the NS equations for each cell [31]. A number of approximations are made, among others a first order Taylor series expansion (linearization) at each cell for the pressure gradients, $\frac{\partial p}{\partial x}$ and $\frac{\partial p}{\partial y}$. The spatial discretization process is outside of the scope of this report. For the full derivation, the reader is referred to the corresponding technical report by Boersma [34].

The next step is to discretize the equations temporally. The implicit scheme described in Versteeg and Malalasekera [31] is followed. This results in

$$\begin{bmatrix} A_x & 0 & B_x \\ 0 & A_y & B_y \\ B_x^T & B_y^T & 0 \end{bmatrix} \begin{pmatrix} \overline{u} \\ \overline{v} \\ \overline{p} \end{pmatrix} = \begin{bmatrix} b_x + \overline{S_x} + u_0 \\ b_y + \overline{S_y} + v_0 \\ b_c \end{bmatrix}, \tag{2-22}$$

where the first two rows represent the discretized terms of Equations (2-6) and (2-7), respectively. The third row represents the continuity equation of Equation (2-5). $A_x$, $A_y$, $u_0$ and $v_0$ originate from the convection and diffusion terms of the $u$- and $v$-momentum equations. $B_x$ and $B_y$ are defined by the mesh layout. $b_x$ and $b_y$ are calculated using the grid layout and the velocity vectors in the grid at the previous time instant, making this system nonlinear. $b_c$ incorporates the boundary conditions, which will be presented later this section. $\overline{S_x}$ and $\overline{S_y}$ include external forces caused by the turbines extracting energy from the flow.

### 2-1-8 Projecting away the continuity equation

In the pursuit of obtaining a control-oriented, state-space model, it is now desired to eliminate the continuity equation given by the third row of Equation (2-22). This is done by defining

$$B_p = \begin{bmatrix} B_x^T & B_y^T \end{bmatrix}^{-1} (-b_c), \tag{2-23}$$

$$Q_p = \text{null}\left(\begin{bmatrix} B_x^T & B_y^T \end{bmatrix}\right), \tag{2-24}$$

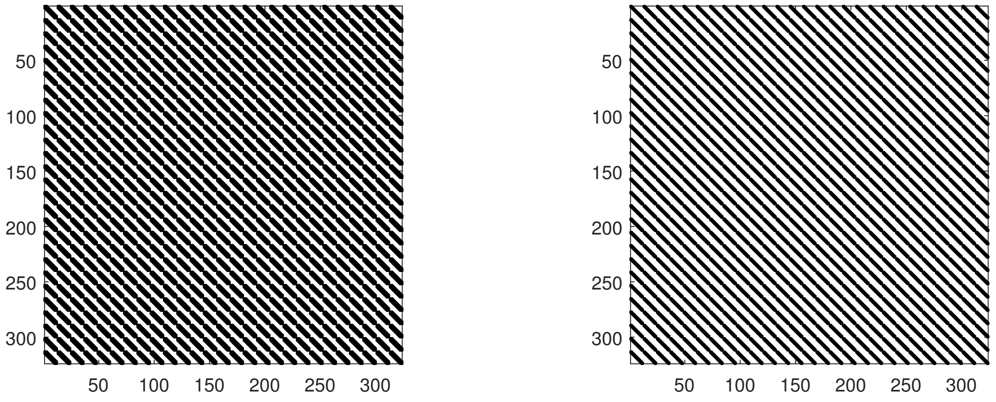
with $B_p$ a solution to the continuity equation, and $Q_p$ the nullspace of $\begin{bmatrix} B_x^T & B_y^T \end{bmatrix}$. Now, by defining

$$\boxed{\begin{bmatrix} \overline{u}_k \\ \overline{v}_k \end{bmatrix} = Q_p \; \alpha_k + B_p,} \tag{2-25}$$

we can reformulate the set of equations given in Equation (2-22) into a state-space form, as

$$\boxed{E_w(\alpha_k)\alpha_{k+1} = A_w \; \alpha_k + B_w(\alpha_k)\beta_{w_k} + S_w(\alpha_k).} \tag{2-26}$$

In these equations, $\alpha_k \in \mathbb{R}^N$ represents the system state vector at time $k$, resulting from the projection of the 2D velocity vectors $[\overline{u}_k; \overline{v}_k] \in \mathbb{R}^M$. Note that the entries of $\alpha_k$ are not physical velocities due to the projection. $N$ is proportional to the number of grid points with a typical value of $10^3$, giving a measure of refinement in flow field estimates. $\beta_{w_k} \in \mathbb{R}^O$ is the system input; a vector with $O$ rows, where $O$ is equal to the number of turbines. This vector is a scaled measure of the axial induction factors $a_k$, representing how much energy is extracted from the wind by each turbine, by $\beta_{w_k} = a_k (1 - a_k)^{-1}$. $\beta_{w_k}$ can be translated into physical turbine settings such as generator torque, blade pitch and yaw. Furthermore, the system matrices are represented by $A_w \in \mathbb{R}^{N \times N}$, $B_p \in \mathbb{R}^M$, $B_w \in \mathbb{R}^{N \times O}$, $E_w \in \mathbb{R}^{N \times N}$, $Q_p \in \mathbb{R}^{M \times N}$ and $S_w \in \mathbb{R}^N$, of which $E_w$, $B_w$ and $S_w$ depend on the state $\alpha_k$. These system matrices are sparse and structured, and can thus be exploited for computational efficiency. An example of sparsity in $E_w(\alpha_k)$ and $A_w$ is displayed in Figure 2-3.



**(a)** Matrix $E_w(\alpha_k) \in \mathbb{R}^{324 \times 324}$

**(b)** Matrix $A_w \in \mathbb{R}^{324 \times 324}$

**Figure 2-3:** Location of nonzero elements (structure) in the system matrices for the WFSim farm model in qLPV form. In this figure, approximately 63% of the entries in $E_w(\alpha_k)$ and about 76% of the entries in $A_w$ are zero.

In this figure, locations of nonzero entries are colored black. About 37% of the entries in $E_w(\alpha_k)$ and 24% of the entries in $A_w$ are nonzero. Please note that $E_w^{-1} A_w$ is a full matrix.

### 2-1-9 Boundary conditions

The NS CFD formulation requires a set of boundary conditions in order for a solution to exist. These boundary conditions are introduced by enforcing a zero flux boundary condition for all but the side of inflow in the grid. For longitudinal velocities, this is formulated as

$$\begin{aligned} n_x \nabla u \left(x = L_x, y\right) &= 0, \\ n_x \nabla u \left(x, y = 0\right) &= 0, \\ n_x \nabla u \left(x, y = L_y\right) &= 0, \end{aligned}$$

where $n_x$ is the unit vector in $x$-direction. For lateral velocities, this is formulated as

$$\begin{aligned} n_y \nabla v \left(x = L_x, y\right) &= 0, \\ n_y \nabla v \left(x, y = 0\right) &= 0, \\ n_y \nabla v \left(x, y = L_y\right) &= 0, \end{aligned}$$

where $n_y$ is the unit vector in $y$-direction. Furthermore, inflow conditions for the remaining side of the grid are to be set. For longitudinal and lateral velocities, this is formulated respectively as

$$\begin{aligned} u(x = 0, y) &= U_\infty, \\ v(x = 0, y) &= V_\infty, \end{aligned}$$

Now, with the predefined freestream flow properties and turbine control settings, one can simulate the flow field iteratively from a set of initial conditions.

### 2-1-10 Software implementation

It is insightful to outline the framework and modules in a practical implementation of the model. The pseudo-code is shown in Algorithm 1.

The simulation options in WFSim include:

1. The timestep ($h$) and total number of steps ($NN$).
2. The domain layout (domain size, number of grid points in long. and lat direction, meshing method (logarithmic or linear), number of turbines, turbine properties).
3. The control inputs, i.e., the turbine control settings.
4. The flow's dynamic viscosity and air density, $\mu$ and $\rho$, respectively.
5. Freestream velocities in longitudinal and lateral direction, $U_\infty$ and $V_\infty$, respectively.
6. The initial pressure and velocity throughout the grid, $\overline{p}_0$, $\overline{u}_0$, $\overline{v}_0$.
7. The choice of whether to solve the problem using the projection matrices (Equation (2-26)) or without (Equation (2-22)).
8. The choice of whether the simulation is dynamic or steady-state.
9. The choice of whether the simulation starts from a steady-state solution or from a uniform (unwaked) flow field.
10. The choice of whether a turbulence model is included, and the value of $l_m$.

Typically, the pressure terms are initiated as $p_0 = 0$. The same assumption is made in the Dynamic Wake Meandering (DWM) model from the Technical University of Denmark (DTU) [35], maintaining good simulation results.

**Algorithm 1:** Pseudo-code WindFarmSimulator (WFSim)

```
inputs : Simulation settings (domain size, meshing, turbine properties, control signal for each
         turbine, freestream flow properties, atmospheric density, dynamic viscosity)
$Wp$ ← Meshing ;                                              // Determine domain and meshing
if Steady state simulation then
    $dt \leftarrow \infty$;   $NN \leftarrow 1$ ;                                  // Steady state simulation
end
$\overline{u}_0 \leftarrow$ ones($Nx$,$Ny$) $\cdot\, U_\infty$ ;                   // Initialize uniform flow field ($\overline{u}$)
$\overline{v}_0 \leftarrow$ ones($Nx$,$Ny$) $\cdot\, V_\infty$ ;                   // Initialize uniform flow field ($\overline{v}$)
$\overline{p}_0 \leftarrow$ ones($Nx$,$Ny$) $\cdot\, p_0$ ;                        // Initialize uniform flow field ($\overline{p}$)
$[B_x, B_y, b_c] \leftarrow$ ComputeTempDiscr ;              // Compute $B_x$, $B_y$ and $b_c$ of eq. (2-22)
if Use projection matrices then
    $[Q_p, B_p] \leftarrow$ ComputeSolutionSpace ;           // Calculate $Q_p$ and $B_p$ of eq. (2-25)
    $\alpha_0 \leftarrow Q_p^{-1}\left(\begin{bmatrix}\overline{u}_0\\ \overline{v}_0\end{bmatrix} - B_p\right)$ ;     // Calculate the initial state by eq. (2-25)
end
for $k \leftarrow 1$ to $NN$ do
    while Solution has not converged do
        if Start at a steady-state solution then
            if $k = 1$ then
                $dt \leftarrow \infty$ ;
            else
                $dt \leftarrow h$ ;
            end
        end
        /* Setup the equality problem of Equation (2-22)                                 */
        ComputeSpatDiscr ;        // Calculate spatial discr.  terms for eq. (2-22) (1/4)
        ComputeDynamics ;              // Calculate dynamical terms for eq. (2-22) (2/4)
        ComputeActuator ;      // Calculate turbine-flow interaction for eq. (2-22) (3/4)
        ComputeBCs ;               // Calculate boundary conditions for eq. (2-22) (4/4)
        /* Compute the solution of Equation (2-22)                                       */
        if Use projection matrices then
            $[E_w, A_w, B_w, S_w] \leftarrow$ ComputeSysMats ; // Calculate system matrices of eq. (2-26)
            $\alpha_k \leftarrow E_w^{-1}\left(A_w \cdot \alpha_{k-1} + B_w \cdot \beta_{w_{k-1}} + S_w\right)$ ;       // Update state by eq. (2-26)
            $\begin{bmatrix}\overline{u}_k\\ \overline{v}_k\end{bmatrix} = Q_p \cdot \alpha_k + B_p$ ;        // Translate state into velocities by eq. (2-25)
        else
            $\begin{pmatrix}\overline{u}_k\\ \overline{v}_k\\ \overline{p}_k\end{pmatrix} \leftarrow \begin{bmatrix}A_x & 0 & B_x\\ 0 & A_y & B_y\\ B_x^T & B_y^T & 0\end{bmatrix}^{-1}\begin{bmatrix}b_x + \overline{S_x} + u_0\\ b_y + \overline{S_y} + v_0\\ b_c\end{bmatrix}$ ;       // Update state by eq. (2-22)
        end
        $[\overline{u}_k, \overline{v}_k, \overline{p}_k] \leftarrow$ UpdateBCs ;   // Update boundary conditions according to section 2-1-9
    end
    if Use control then
        $[\beta_{w_k}, \gamma_k] \leftarrow$ ComputeInputs ;               // Compute turbine control settings
    end
end
```

### 2-1-11 Benefits and challenges in application

Because of the state-space formulation of the flow model, WFSim can be applied for dynamic control in a closed-loop setting using a state observer. Furthermore, the model is flexible for various wind farm layouts, and has a higher fidelity than the typical heuristic static flow model such as Jensen and FLORIS.

However, due to its increased fidelity it is computationally significantly more heavy. A typical size of state $\alpha_k$ is of the order $10^3$. Designing observers and predictive control algorithms that satisfy the time restrictions for real-time operation are serious challenges, and are currently active topics of research. The idea is to exploit the sparse structure of the system matrices, matrix transformations, parallelization of computations, and apply state of the art algorithms from literature aimed at large order systems.

## 2-2 The SOWFA flow model

The high fidelity simulation model used for validation is Simulator fOr Wind Farm Applications (SOWFA) from the National Renewable Energy Laboratory (NREL). This wind farm model provides highly accurate flow data at a fraction of the cost of field tests. Churchfield *et al.* [36] give an overview of previous research, and discuss the simulation methodology more thoroughly. Rather, in this document a short introduction to the model is given.

SOWFA predicts the velocity vectors in a three dimensional grid using a large-eddy simulation (LES) method. It relies on the 3D incompressible NS equations for a steady or unsteady flow field, accounting for bouyancy (based on Boussinesq approximation [15]) and Coriolis effects [37]. LES methods resolve larger scale dynamics directly, but employ a subgrid-scale model for small eddy dynamics to reduce computational cost.

For rotor modeling it employs the actuator line model (ALM), which is a more sophisticated version of the ADM. In the ALM, the rotor blades are discretized spatially along their radial lines, where lift and drag forces are determined based on the incoming flow angle, flow velocity, and blade (airfoil) geometry [15].

For turbine modeling the Fatigue, Aerodynamics, Structures and Turbulence (FAST) model [23] from NREL is implemented. This model calculates, among others: the power production of the turbine, the blade forces on the flow, the structural loading on the turbine, and the dynamics (position, speed, acceleration) of a number of different turbine components [12].

SOWFA has been used on multiple occasions for lower fidelity model validation, steady-state controller testing, and to study the aerodynamics in wind farms (e.g., in [12, 15, 16, 20, 38]). While SOWFA is assumed to be very close to the truth in many simulations, its validation is still an ongoing process. Currently, field tests have shown accurate simulation results for the first 5 rows of turbines [20]. As only a two turbine case is used for simulations in this work, results are deemed reliable. For a full description of the flow equations, please see the article by Churchfield *et al.* [36], which also gives a more in-depth description of LES solvers in general.

## 2-3 Model validation of WFSim with SOWFA

It is worthwhile to consider the accuracy of WFSim when compared to higher fidelity models and field tests, as this will give a notion of the model's reliability, and furthermore of the degree of correction that is necessary for closed-loop control. With the limited availability of field test data, no experimental data is employed for validation at this point.

In the paper by Boersma *et al.* [28], the flow model WFSim (excluding the mixing length turbulence model) is validated according to actuator disk theory, which predicts the flow behavior in streamwise direction in the presence of a turbine extracting energy from the wind. Two important equations in this validation process are Equations (2-17) and (2-18). Good results were shown. However, actuator disk theory is an unrealistic, very simplified representation of reality, and often insufficient for higher fidelity modeling. For this reason, WFSim is further validated using high fidelity simulation data in this work.

## 2-3-1 Simulation setup

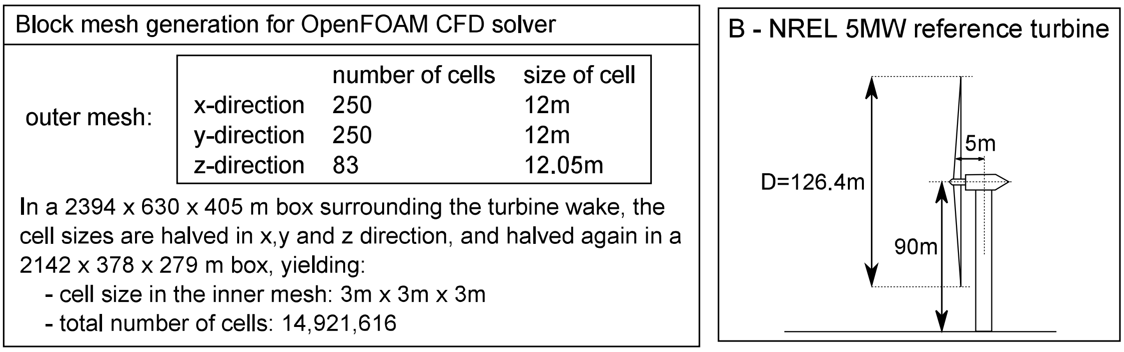
A number of SOWFA simulation datasets are readily available in the online repository.[5] WFSim will be simulated under settings that represent those of the SOWFA simulation as accurately as possible. This set of optimal settings, the "mapping", is to be found first.

A 2-turbine case SOWFA simulation that excludes a precursor simulation[6] is used for model validation of WFSim, expecting improved validation results compared to datasets that include a precursor. Also this dataset is available on NREL's repository.[7] This dataset includes both horizontal slices at hub height and vertical slices intersecting with both turbines. The simulated domain is of size 3000 by 3000 by 1000 meters, and contains two identical NREL 5-MW wind turbines at $(1226.3, 1342.0)$ m and $(1773.7, 1658.0)$ m, respectively. A number of properties of the NREL 5-MW wind turbine are listed in Table 2-1.

**Table 2-1:** Properties of the NREL 5-MW baseline wind turbine. [39]

| | |
|---|---|
| Power rating | 5 MW |
| Rotor configuration | Upwind, 3 blades |
| Control configuration | Variable speed, collective pitch |
| Drivetrain properties | High speed, multiple-stage gearbox |
| Rotor diameter ($D$), hub height | 126 m, 90 m |
| Cut-in, rated, cut-out wind speed | 3 m/s, 11.4 m/s, 25 m/s |
| Cut-in, rated rotor speed | 6.9 rpm, 12.1 rpm |
| Rotor, nacelle, tower mass | $1.1 \cdot 10^5$ kg, $2.4 \cdot 10^5$ kg, $3.5 \cdot 10^5$ kg |

The incoming wind flow is aligned orthogonal to both rotor planes. The CFD solver and turbine properties are displayed in Figure 2-4, and the mesh layout is displayed in Figure 2-5.



Block mesh generation for OpenFOAM CFD solver

outer mesh:

| | number of cells | size of cell |
|---|---|---|
| x-direction | 250 | 12m |
| y-direction | 250 | 12m |
| z-direction | 83 | 12.05m |

In a 2394 x 630 x 405 m box surrounding the turbine wake, the cell sizes are halved in x,y and z direction, and halved again in a 2142 x 378 x 279 m box, yielding:

- cell size in the inner mesh: 3m x 3m x 3m
- total number of cells: 14,921,616

**Figure 2-4:** SOWFA simulation setup: CFD solver settings and turbine properties. [12]

[5] Datasets can be downloaded from NREL's repository at http://wind.nrel.gov/public/ssc/.

[6] Traditionally, high fidelity SOWFA simulations are initialized from a turbulent flow field, created by performing a so-called "precursor simulation". In this precursor, turbulent forces in the flow develop in the absence of turbines, eventually resulting in a turbulent, quasi-steady flow field. This developed flow field is then used as an initial field for the actual simulation, in which the turbine structures are introduced. WFSim supports no such feature yet, and furthermore only includes a very minimalistic turbulence model. Rather, in WFSim, simulations are initialized from a fully homogeneous flow field. Hence, using a dataset for model validation that includes a precursor would complicate this process.

[7] URL: http://wind.nrel.gov/public/ssc/sampleForPODexcitationPRBSpositiveNoPrecursor/.

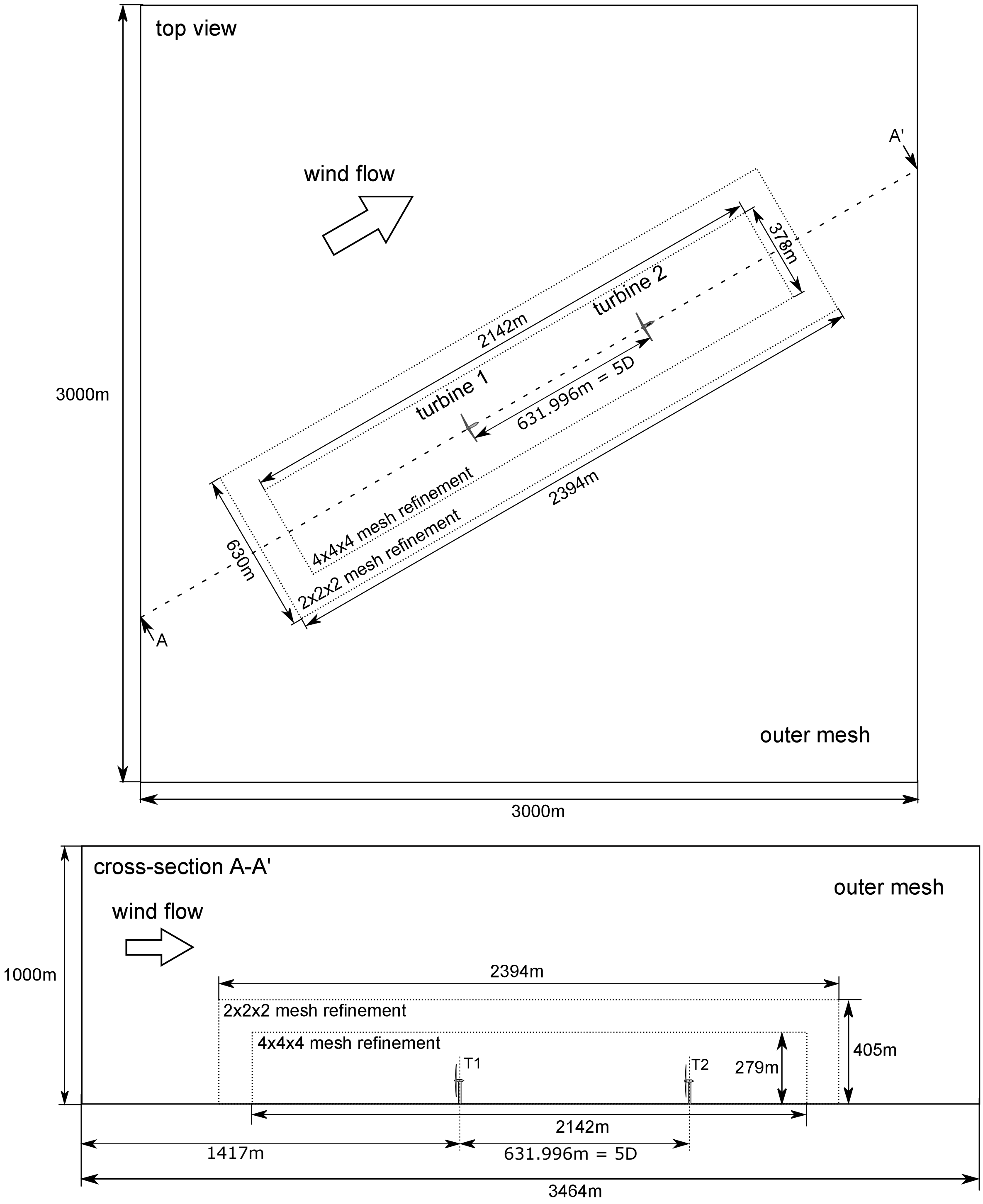


**Figure 2-5:** SOWFA simulation setup: a two turbine case with three different mesh resolutions: from coarse to fine near the turbines. The domain size is $3000$ by $3000$ by $1000$ m, with two identical turbines at $(1226.3, 1342.0)$ m and $(1773.7, 1658.0)$ m, each with rotor diameter $D = 126.3992$ m. For the simulation without precursor, datasets are available containing horizontal slices at hub height, and vertical slices intersecting both turbines (cross-section $A - A'$). Figure taken and modified from Gebraad [12].

In this simulation, the first turbine periodically excites the downstream flow by adjusting the collective pitch angle according to a pseudo-random binary sequence (PRBS) signal, as shown in Figure 2-6. The yaw angle and generator torque of the first turbine do not change. The second turbine is operating at constant pitch angle $\beta = 0°$, yaw $\gamma = 0°$, and generator torque.

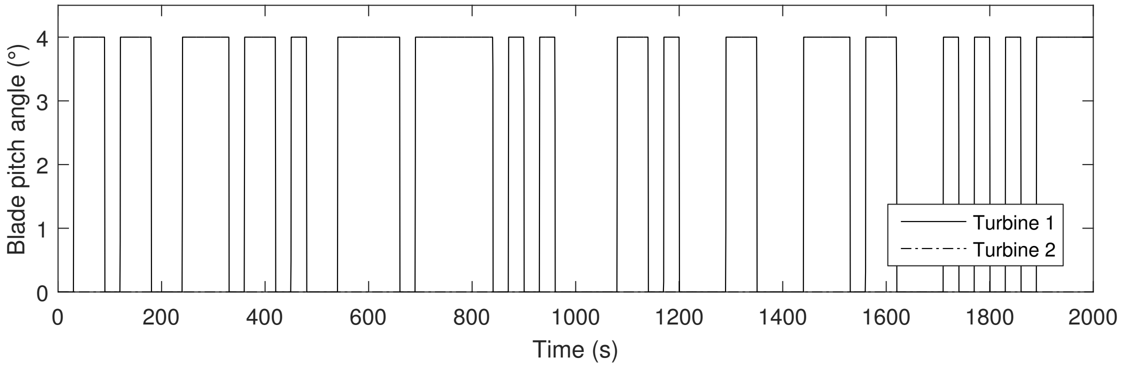


**Figure 2-6:** Collective pitch angle for turbine 1 in the SOWFA simulation dataset. Turbine 1 follows a PRBS signal between $0°$ and $4°$, while turbine 2 is kept at a constant pitch angle of $0°$.

Furthermore, SOWFA is simulated at 50 Hz, but the data is resampled at 1 Hz. WFSim will be simulated at 1 Hz.

## 2-3-2 Mapping SOWFA to WFSim

For validation, WFSim should be simulated under similar conditions as the high fidelity SOWFA model. To allow this, the high fidelity dataset is to be analyzed and manipulated first.

### Domain size and meshing

As can be seen in Figure 2-5, the domain is of large size (3000 by 3000 m), with the area of interest only being a small portion of this. While a simulation with identical meshing is possible in WFSim, it would result in an unnecessarily high computational cost. Furthermore, the inflow in SOWFA is under a 30° angle from the horizontal axis, as seen in the figure. Preferred is a horizontal or vertical inflow.

For these reasons, the domain is first rotated clockwise by 30°, then translated such that turbine 1 is at $(500, 400)$ m and turbine 2 at $(500, 1031.9960)$ m, and finally the domain is cropped to 1000 by 2000 m. The distance between the two turbines remains unaltered at 5 rotor diameters $D$, and the mesh extends to about $8D$ behind the second turbine to simulate the relevant wake dynamics. Furthermore, the domain is remeshed using linear interpolation onto the meshes produced in WFSim, once at 50 by 100 points ($N = 4.6 \cdot 10^3$), and once at 25 by 50 points ($N = 1.0 \cdot 10^3$) according to the 'exponential meshing' algorithm. The different sampling qualities for the raw data, for SOWFA remeshed at 50 by 100 mesh points, and SOWFA remeshed at 25 by 50 mesh points, are compared in Figure 2-7. The loss in information is clearly distinguishable between the coarser and finer mesh to be used in WFSim. Finally, the new simulation settings are summarized in Figure 2-8.

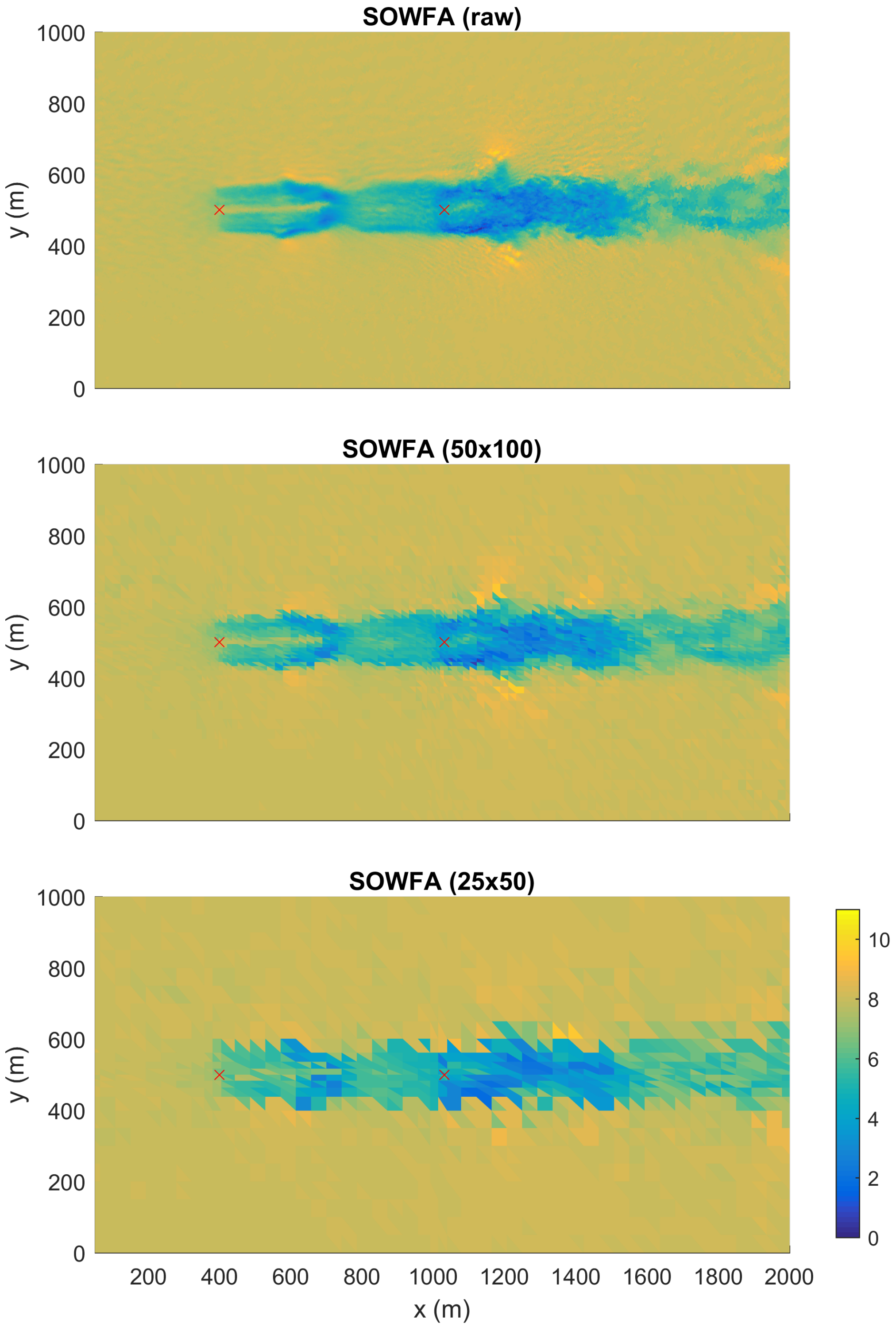


**Figure 2-7:** Comparison of meshing resolution obtained by linearly interpolating the raw dataset (high fidelity data from SOWFA) onto the 25x50 and 50x100 meshes used in WFSim.

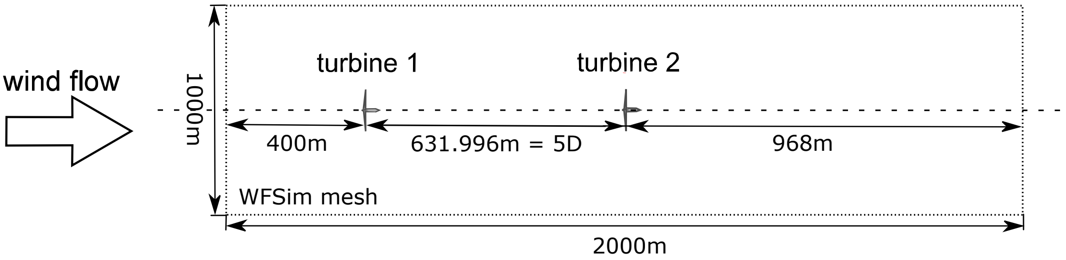


**Figure 2-8:** Validation simulation setup: a two turbine case. The domain is a horizontal plane at hub height with size $1000$ by $2000$ m, containing two identical turbines at $(400, 500)$ m and $(1031.9960, 500)$ m, with rotor diameter $D = 126.3992$ m. Figure taken and modified from Gebraad [12].

### Turbine operating settings

A second nontrivial task is mapping the turbine operating settings in SOWFA to that of WFSim. Notably, SOWFA uses the ALM for rotor modeling, relying on physical quantities such as the yaw angle, generator torque and individual blade pitch angles. On the other hand, WFSim uses a simplified ADM in which only an axial induction factor $a$ and yaw angle $\gamma$ for each turbine is defined. Therefore, a mapping has to be established between the physical inputs in SOWFA and the theoretical axial induction factors in WFSim.

Conceptually, the idea is that turbine 1 switches between different axial induction factors, synchronous to the switches in pitch angle displayed in Figure 2-6. As turbine 2 is operating at constant control settings, it should operate at a constant axial induction factor.

A number of approaches can be followed to obtain axial induction factors from the high fidelity dataset, most importantly:

- Using the $C_P - C_T$ curve, and then relating $C_T$ to $a$. See section 3 of [40] for an example.
- Calculating the mean streamwise velocities (wake centerlines) through the turbines from SOWFA data, and then tuning the axial induction factor for the ADM in WFSim to obtain minimal root mean square (RMS) error. Namely, actuator disk theory predicts the velocity in longitudinal direction to be inverted hysteresis functions centered at the turbine rotors, by

$$U(\Delta x) = U_\infty \cdot 2a \left[ \frac{1}{2} + \frac{1}{\pi} \arctan \left( \frac{2.0 \cdot \alpha_t \cdot \Delta x}{D} \right) \right], \tag{2-27}$$

  with $\alpha_t \approx 1$ a tuning parameter, and $\Delta x$ the longitudinal displacement from the rotor disk [28, 41].
- Translating turbine dynamics and structural loading to an equivalent axial induction factor by applying actuator disk theory, blade element theory, or basic Newtonian mechanics.

The option followed in this work is the latter: the tower fore-aft bending moment $M_z$ is translated back into an axial force on the rotor blade according to $M_z = F_T \cdot h_z$, with $h_z$ the turbine hub height. Its implementation is very straight-forward, robust, and does not require airfoil-specific $C_P - C_T$ curves. From actuator disk theory [28], the axial force at the rotor plane $F_T$ is

$$F_T = \frac{1}{2} \rho U_\infty^2 \left( \frac{1}{4} \pi D^2 \right) C_T. \tag{2-28}$$

Substituting $C_T = 4a(1-a)$ from actuator disk theory, and Equation (2-17) to replace dependency on the freestream velocity $U_\infty$ with dependency on the velocity at the rotor disk $U_D$, yields

$$F_T = 2\rho\frac{a}{(1-a)}U_D^2\left(\frac{1}{4}\pi D^2\right) = 2\rho\beta_w U_D^2\left(\frac{1}{4}\pi D^2\right). \tag{2-29}$$

From this equation, $\beta_w$ can directly be obtained if $U_D$ is known.

Two velocities near the hub $U_D$ are reported in SOWFA: one from the turbine submodel FAST and one from the regular flow model computed at each grid coordinate. However, neither give a reliable estimation of the mean wind speed at hub height. The values from FAST are unreliable, because no turbine hub is modeled, and a ’funnel effect’ is induced [41]. Furthermore, SOWFA’s flow data is also unreliable due to local blade effects that disturb the flow, resulting in large flow fluctuations at similar locations on the rotor. For this reason, a lateral velocity average from rotor end to rotor end is taken to calculate an estimate for $U_D$. The area laterally spans the same distance as one rotor diameter $1D$.

Now, the input vectors $\beta_w$ can actually be determined. The fore-aft bending moment data is low-pass filtered (LPF), after which Equation (2-29) is reformulated, resulting in

$$\beta_w = \frac{a}{1-a} = F_T\left(2\rho U_D^2\left(\frac{1}{4}\pi D^2\right)\right)^{-1} = \frac{M_z}{h_z}\left(2\rho U_D^2\left(\frac{1}{4}\pi D^2\right)\right)^{-1}, \tag{2-30}$$

From which the estimates are obtained as presented in Figure 2-9. $\beta_w$ switches between 0.153 and 0.278 for the first turbine, while turbine 2 is operated at a constant $\beta_w$ of 0.302. The switching in the first turbine is synchronous to the changes in the collective pitch angle, as previously shown in Figure 2-6, and also suggested by the raw and LPF filtered data.

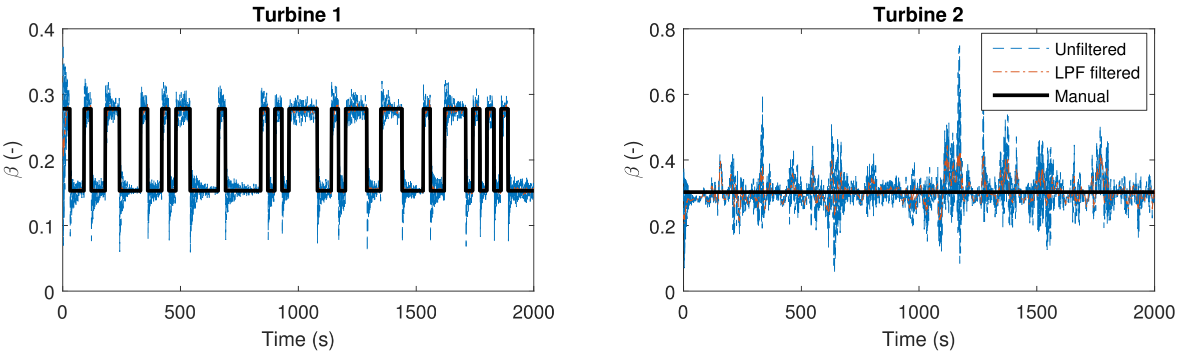


**Figure 2-9:** Estimated input signal $\beta_{w_k}$ for the SOWFA dataset. For turbine 1, $\beta_w$ switches between 0.153 and 0.278, synchronous to the blade pitch angle signal (Figure 2-6). For turbine 2, $\beta_w$ is 0.302.

While the estimate for the first turbine looks accurate, there is much more fluctuation in $\beta_w$ for the second turbine. However, it is still opted to choose a constant $\beta_w$, following physical intuition. For clarity, the $\beta_w$ signal to be implemented in WFSim will be ‘Manual’, obtained by manually fitting the unfiltered and filtered signals with a signal switching between a high and low setting. As manual tuning and fitting will introduce some error in the input signal, this will set a more realistic simulation scenario – the mapping from physical control settings to a theoretical value is ambiguous, especially when applied to real wind farms.

### Inflow, turbulence and atmospheric properties

As no precursor simulation is present, the inflow is nearly uniform in the high fidelity dataset. The longitudinal inflow component has a mean value of 8.0 m/s and a standard deviation increasing from 0.02 m/s to about 0.14 m/s at the end of the simulation. The lateral component has a mean value of 0.02 m/s, and standard deviation fluctuating between 0.01 and 0.04 m/s. In response, WFSim will be simulated with a fully uniform inflow with longitudinal components $U_\infty = 8.0$ m/s and lateral components $V_\infty = 0.0$ m/s.

WFSim includes a very simple turbulence model only for the streamwise flow. On the other hand, SOWFA includes turbulence modeling in all 3 dimensions at high resolution. In the pursuit of matching the two models, the eddy length scale $l_m$ is tuned according to the SOWFA wake model. This results in $l_m = 0.60$ m, which improves modeling results compared to the absence of a turbulence submodel. Typically, the value of $l_m$ is proportional to the amount of mixing in the flow, and thereby wake recovery.

Concerning the atmospheric properties, SOWFA does not directly output a dynamic viscosity or a flow density, as they are calculated dynamically according to thermodynamic laws. The flow density at an arbitrary instant of the SOWFA dataset is taken and inserted in WFSim, assuming incompressibility of the flow: $\rho = 1.2231$ kg m$^{-3}$. The dynamic viscosity is assumed $0.18 \cdot 10^{-5}$ Pa s, a standard value for air at room temperature.

## 2-3-3 Validation results

WFSim is now simulated under the settings matching SOWFA as defined in Section 2-3-2. WFSim outputs the velocities in longitudinal and lateral direction at their corresponding mesh coordinates for each time instant. The data is analyzed next. Please note that quantitative and qualitative measures of validity will be introduced more formally, along with observer results, in Section 4-2. Results presented in this section are both for the 50x100 meshing and for the 25x50 meshing.

**Flow fields** Firstly, contour plots that display the complete flow field for both SOWFA and WFSim at various time instants are displayed in Figure 2-10. Contour plots of the flow in lateral direction can be found in Appendix B, as these are of secondary importance. A number of things can be noted from Figure 2-10. Firstly, it can clearly be seen how SOWFA contains much stochasticity in its flow due to higher fidelity turbulence modeling and the vertical dimension (3D vs. 2D). WFSim does not capture all the dynamics that SOWFA does.

Secondly, the wake location and structure differs. The wake is much wider in WFSim than in SOWFA, possibly due to the absence of turbulence modeling in lateral direction. Also, behind the rotor, two wakes form in SOWFA due to the presence of the turbine hub, which is not modeled in WFSim.[8] Furthermore, WFSim models a symmetric wake as the model does not account for wake meandering, wake skewing or wake deflection due to rotor rotation.

Thirdly, the velocity in the wake has a lower minimum compared to WFSim in a number of time instances. These results suggest that $\beta_w$ is underestimated for certain cases, and may need to be corrected for.

[8] Note that research is currently ongoing to improve this based on work by Annoni and Seiler [42].

Fourthly, due to the simplified turbulence model, changes in WFSim are mainly due to changes in the control settings $\beta_w$, which give rise to small changes in the wake. Therefore, after the wake has fully been developed, it is difficult to notice any real changes in the field for WFSim.

Fifthly, looking at the estimation error plots, it is seen that most error is found around the second turbine and its wake downstream. This is also as expected from Figure 2-9, in which it was shown that there was a significant variation on the estimated axial induction factor for turbine 2. This flow is very turbulent and the dynamics are complicated, for which WFSim is not sophisticated enough to capture.

Furthermore, the resemblance in flow build-up can be seen in front of turbine 1 for both SOWFA and WFSim.

**Maximum and mean error** Secondly, from the flow fields, the maximum and mean error between WFSim and SOWFA for both lateral and longitudinal wind speeds can be calculated, as displayed in Figure 2-11. However, this figure is relatively deceiving, because maximum errors remain large due to the near wake region behind the turbines containing complex flow dynamics, while the remainder of the flow may be estimated correctly. Errors originate from the inferior rotor submodel included in WFSim that neglects the near wake region, leaving large estimation errors just downstream of the turbines. The average estimation error remains around 0.39 m/s after the first 200 seconds for both the 50x100 and 25x50 WFSim simulations.

**Wake centerlines** Thirdly, the mean wake centerline at various time instants is calculated and displayed in Figure 2-13. This mean velocity is defined as the laterally-averaged streamwise flow velocity throughout the simulation domain, intersecting with the two turbines, according to Figure 2-12. From Figure 2-13, a number of things should be noted. Firstly, the flow is generally well predicted by WFSim in front of, and around the first turbine ($x = 400$ m). This is further consolidated by Figure 2-9, in which a signal with relatively low noise was shown for $\beta_w$ at the first turbine. The main reason for this is the (nearly) uniform inflow, showing much agreement with actuator disk theory. On the other hand, the wake centerline is predicted less well for the second turbine ($x = 1032$ m). The flow upstream of turbine 2 contains much more turbulence and is far from uniform due to energy extraction by the first turbine, inducing larger errors when compared to the simplified ADM implemented in WFSim. This is consolidated by Figure 2-9, in which the signal used to determine $\beta_w$ for turbine 2 contains much more noise.

Furthermore, SOWFA predicts more wake recovery than WFSim does. This is expected to be due to the lack of accurate turbulence models in WFSim, thereby neglecting effects that would otherwise promote wake recovery. As mentioned in Section 2-1-5, convection and diffusion are not included in the current turbulence model.

Also, the corresponding root mean square (RMS) error, variance accounted for (VAF) values, and QOF values are displayed in Table 2-2, confirming the aforementioned statements quantitatively.[9] Additionally, in all 6 time instants, the higher resolution model (WFSim 50x100 vs. WFSim 25x50) has a smaller average RMS, a higher VAF, and a higher QOF for time $t > 1$ s, as seen in Table 2-2.

[9] The definitions of the RMS error, VAF and QOF will be introduced more formally in Section 4-2.

**Table 2-2:** RMS error, VAF, and QOF between the true mean wake centerline obtained from SOWFA data and the estimated mean wake centerline obtained from WFSim.

**(a)** WFSim 50x100.

| **Time (s)** | **RMS (m/s)** | **VAF (%)** | **QOF (%)** |
|---|---|---|---|
| 1 | $5.78 \cdot 10^{-2}$ | 16.8 | 100 |
| 200 | $4.60 \cdot 10^{-1}$ | 85.0 | 99.4 |
| 500 | $7.11 \cdot 10^{-1}$ | 73.5 | 98.5 |
| 1000 | $5.76 \cdot 10^{-1}$ | 82.1 | 99.0 |
| 1500 | $6.69 \cdot 10^{-1}$ | 77.5 | 98.6 |
| 1999 | $3.72 \cdot 10^{-1}$ | 92.0 | 99.6 |

**(b)** WFSim 25x50.

| **Time (s)** | **RMS (m/s)** | **VAF (%)** | **QOF (%)** |
|---|---|---|---|
| 1 | $4.61 \cdot 10^{-2}$ | 17.2 | 100 |
| 200 | $4.91 \cdot 10^{-1}$ | 85.3 | 99.4 |
| 500 | $8.36 \cdot 10^{-1}$ | 64.6 | 98.1 |
| 1000 | $6.42 \cdot 10^{-1}$ | 77.9 | 98.8 |
| 1500 | $7.85 \cdot 10^{-1}$ | 68.8 | 98.3 |
| 1999 | $4.99 \cdot 10^{-1}$ | 88.6 | 99.3 |

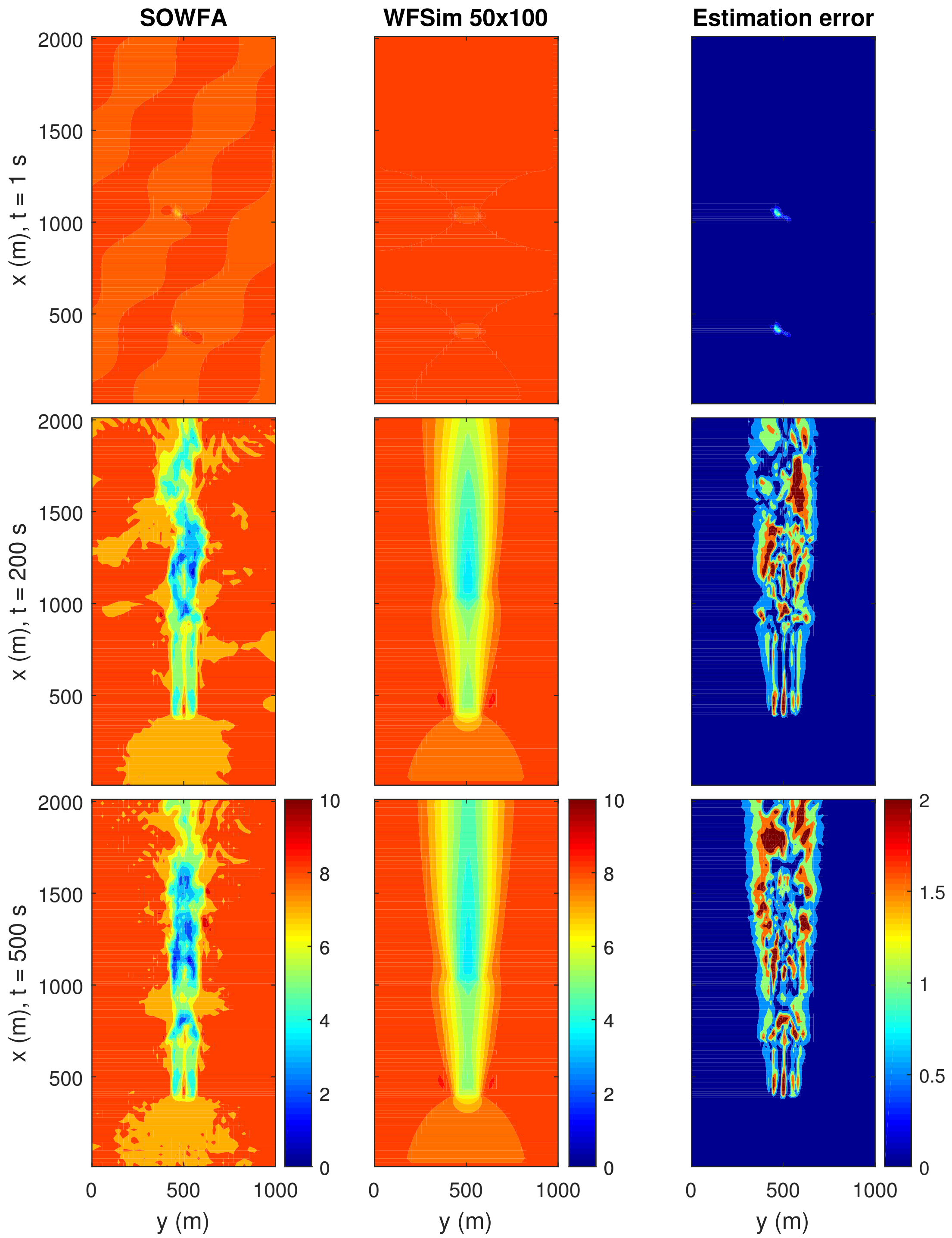
**Wake centerline dynamics** Finally, the mean wake centerlines over the entire simulation time can be summarized in a single figure by plotting the spatial coordinate against the temporal coordinate, as presented in Figure 2-14. These images show the dynamics of the mean wake centerline, its evolution and its resemblance with SOWFA directly. Each vertical segment corresponds to the mean wake centerline at one specific time instant. The irregularly dashed black line at $x = 400$ m indicates the control setting of the first turbine, as denoted in Figure 2-9. At the locations of the black lines $\beta_w = 0278$, and for the gaps $\beta_w = 0.153$. The regularly dotted line indicates the second turbine, operating at a constant setting $\beta_w = 0.302$. This figure clearly shows the effect of the change in pitch angle of turbine 1 on the flow wakes and the turbine's energy extraction. From this figure, it becomes very clear that WFSim, although predicting the overall flow dynamics well, misses a lot of stochasticity and flow characteristics seen in SOWFA – WFSim "blends out" the flow. Also, wake recovery is underestimated in WFSim.

Notably, the wake is deeper when the turbine is operating at zero blade pitch angle ($\beta_w = 0.278$), and this effect is seen both in WFSim and in SOWFA. This figure furthermore confirms the claim that the wake is deeper in SOWFA, and $\beta_w$ appears to be underestimated in some situations. Additionally, the flow propagates and the wakes develop in approximately the same time, as can be seen in the first 200 seconds (most left part) of the figures.

**Conclusion** In conclusion, WFSim provides accurate predictions of the flow for a uniform inflow case in a single turbine simulation. Adding a second turbine will induce more complicated flow dynamics due to wake development and flow-turbine interactions. WFSim is not sophisticated enough to model all these interactions accurately, and only predicts an averaged, smoothened flow. This averaged flow does give very satisfactory results, when looking at Figure 2-13. However, it is desired to capture more dynamics.

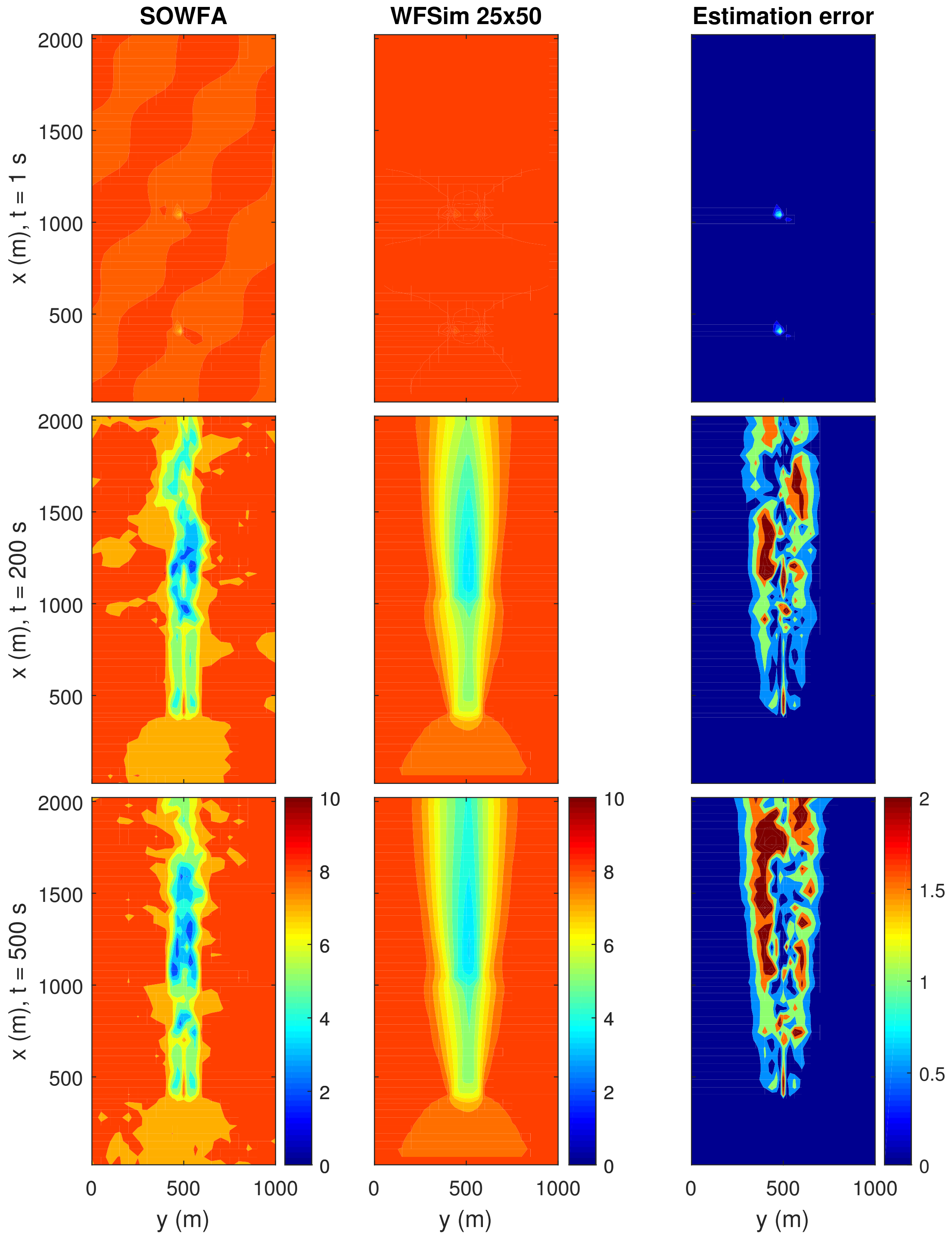
Due to the simplified turbulence and rotor model, the modeled flow lacks specifically in the near wake region and in wake recovery, as seen in Figures 2-10, 2-13 and 2-14. For example, turbine effects such as the presence of the hub are not modeled. Furthermore, the wake width is overestimated, possibly due to the absence of a turbulence model in lateral direction.

An observer would potentially improve the resemblance of this model with SOWFA and, if successful, too with field test data. Noticeably, the near wake effects, the wake width and the wake recovery are critical points for further model improvement and observer performance assessment.



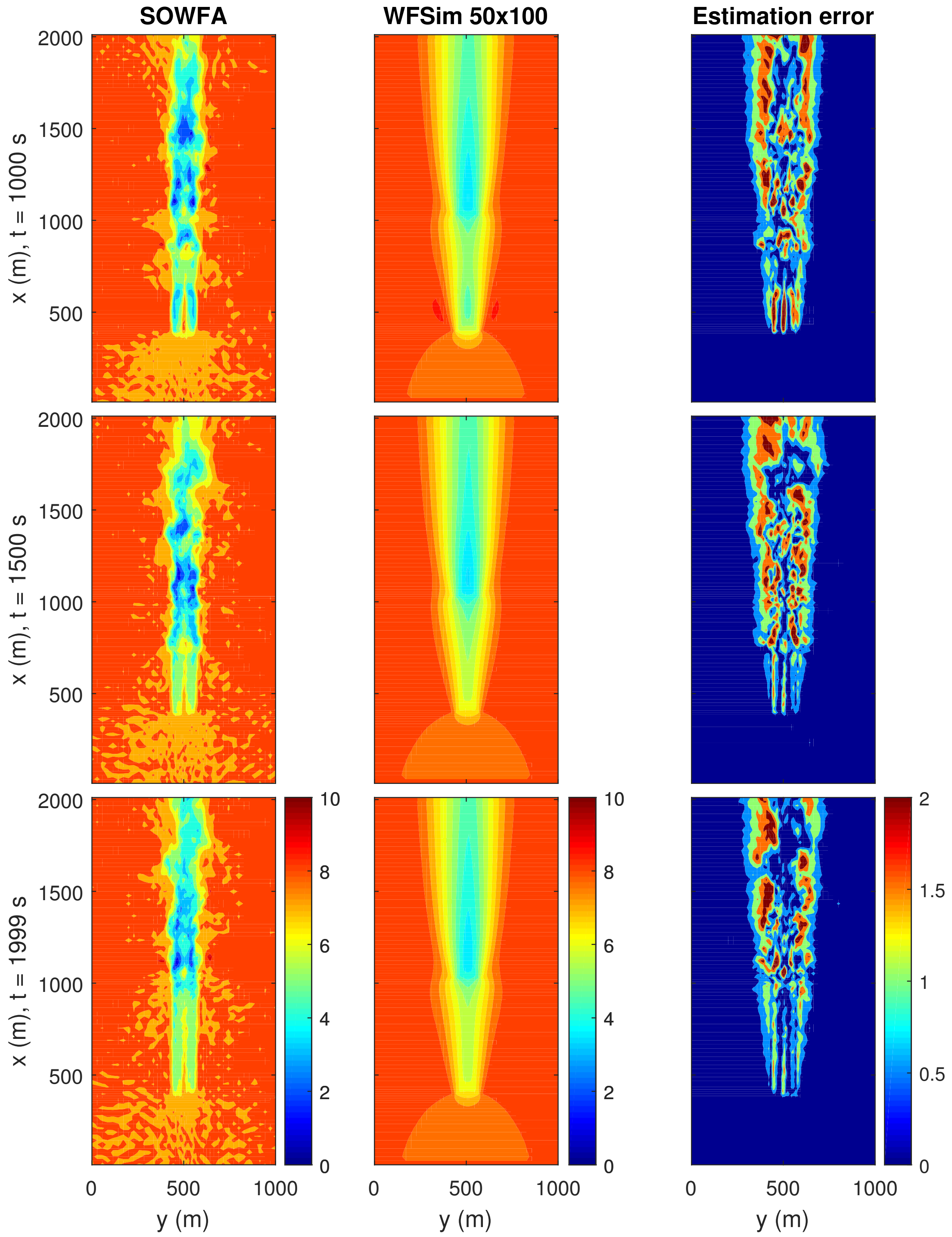
**(a)** Longitudinal flow velocity for 50x100 meshing at $t = 1, 200, 500$ s.

**Figure 2-10:** Snapshots of the longitudinal flow velocity (m/s) throughout the grid for various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s for the WFSim validation case. The raw data is shown on the left using SOWFA data resampled at a 50x100 and 25x50 meshing, respectively. WFSim is simulated under a 50x100 and a 25x50 meshing.



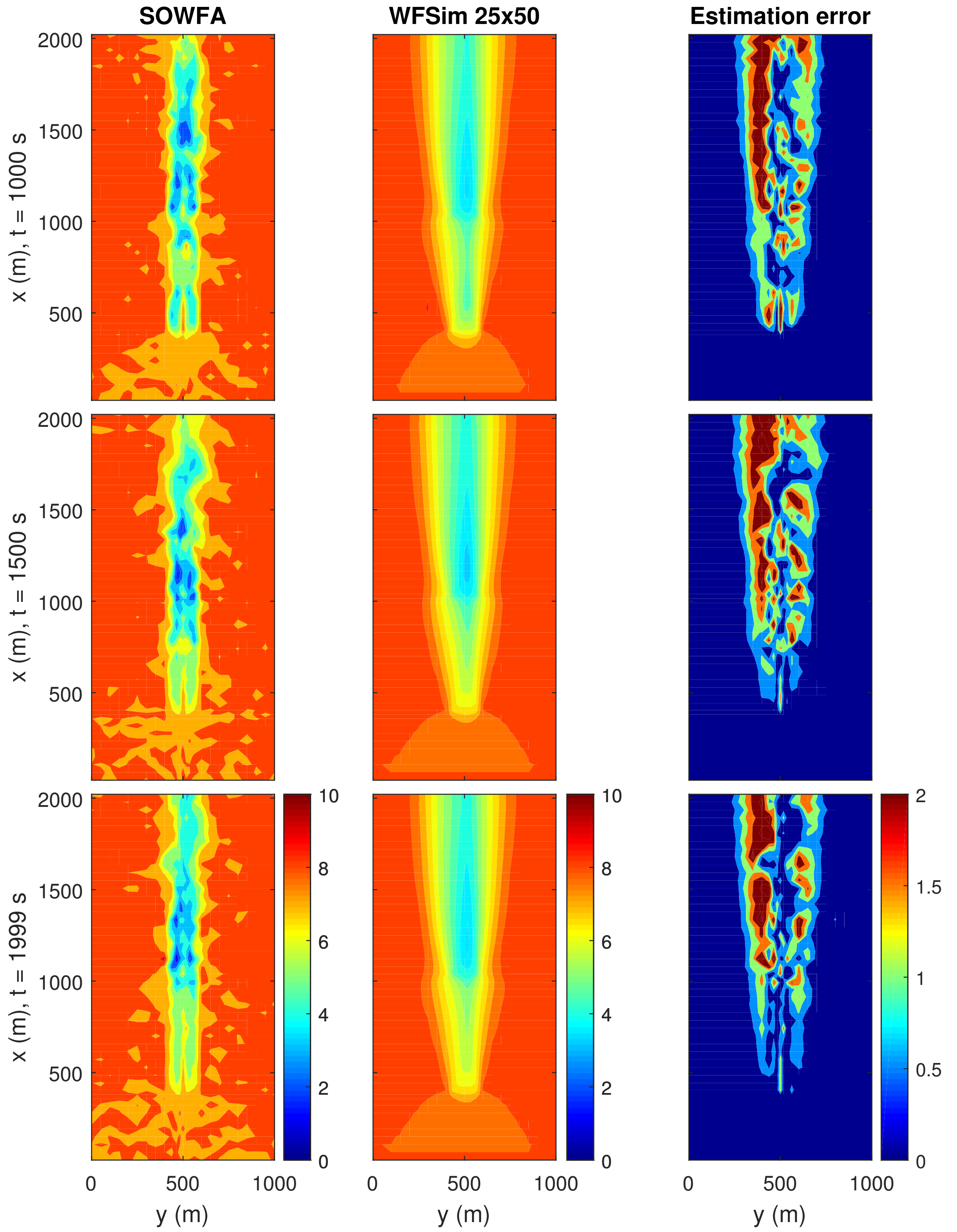
**(b)** Longitudinal flow velocity for 25x50 meshing at $t = 1, 200, 500$ s.

**Figure 2-10:** Snapshots of the longitudinal flow velocity (m/s) throughout the grid for various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s for the WFSim validation case. The raw data is shown on the left using SOWFA data resampled at a 50x100 and 25x50 meshing, respectively. WFSim is simulated under a 50x100 and a 25x50 meshing.



**(c)** Longitudinal flow velocity for 50x100 meshing at $t = 1000, 1500, 1999$ s.

**Figure 2-10:** Snapshots of the longitudinal flow velocity (m/s) throughout the grid for various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s for the WFSim validation case. The raw data is shown on the left using SOWFA data resampled at a 50x100 and 25x50 meshing, respectively. WFSim is simulated under a 50x100 and a 25x50 meshing.



**(d)** Longitudinal flow velocity for 25x50 meshing at $t = 1000, 1500, 1999$ s.

**Figure 2-10:** Snapshots of the longitudinal flow velocity (m/s) throughout the grid for various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s for the WFSim validation case. The raw data is shown on the left using SOWFA data resampled at a 50x100 and 25x50 meshing, respectively. WFSim is simulated under a 50x100 and a 25x50 meshing.

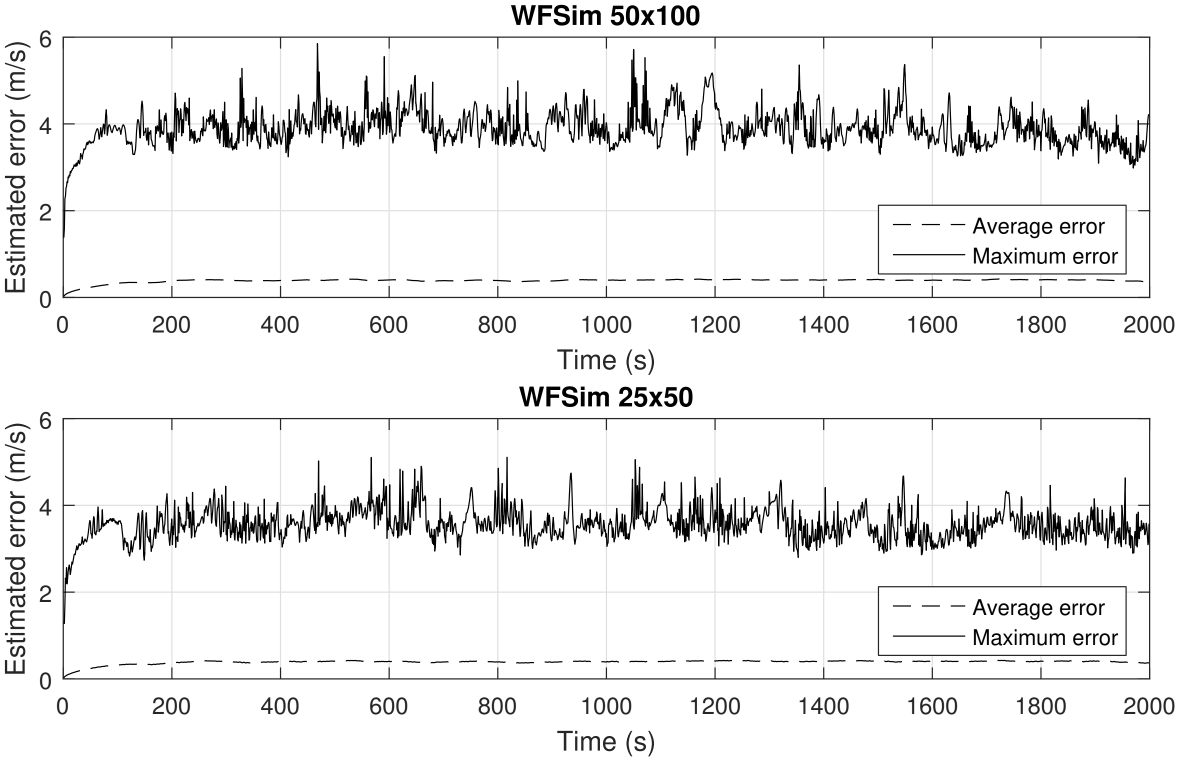


**Figure 2-11:** Error between SOWFA and WFSim over time (m/s) for both the 50x100 meshing and the 25x50 meshing. The errors in this figure are calculated from both lateral and longitudinal velocities, spanning the entire flow field.

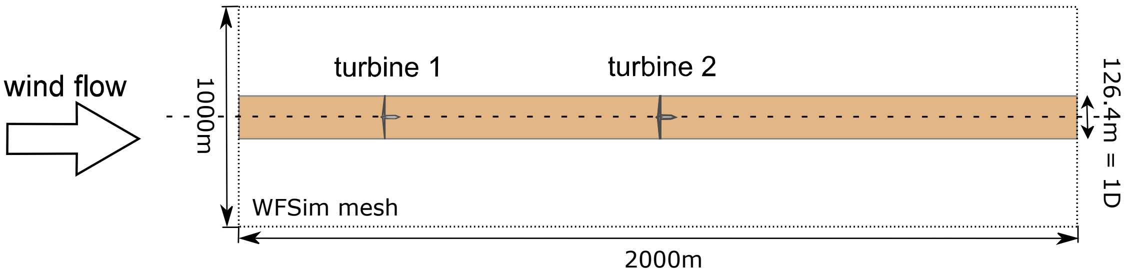


**Figure 2-12:** Extracted area used to determine a mean longitudinal wake centerline velocity profile. Figure taken and modified from Gebraad [12].

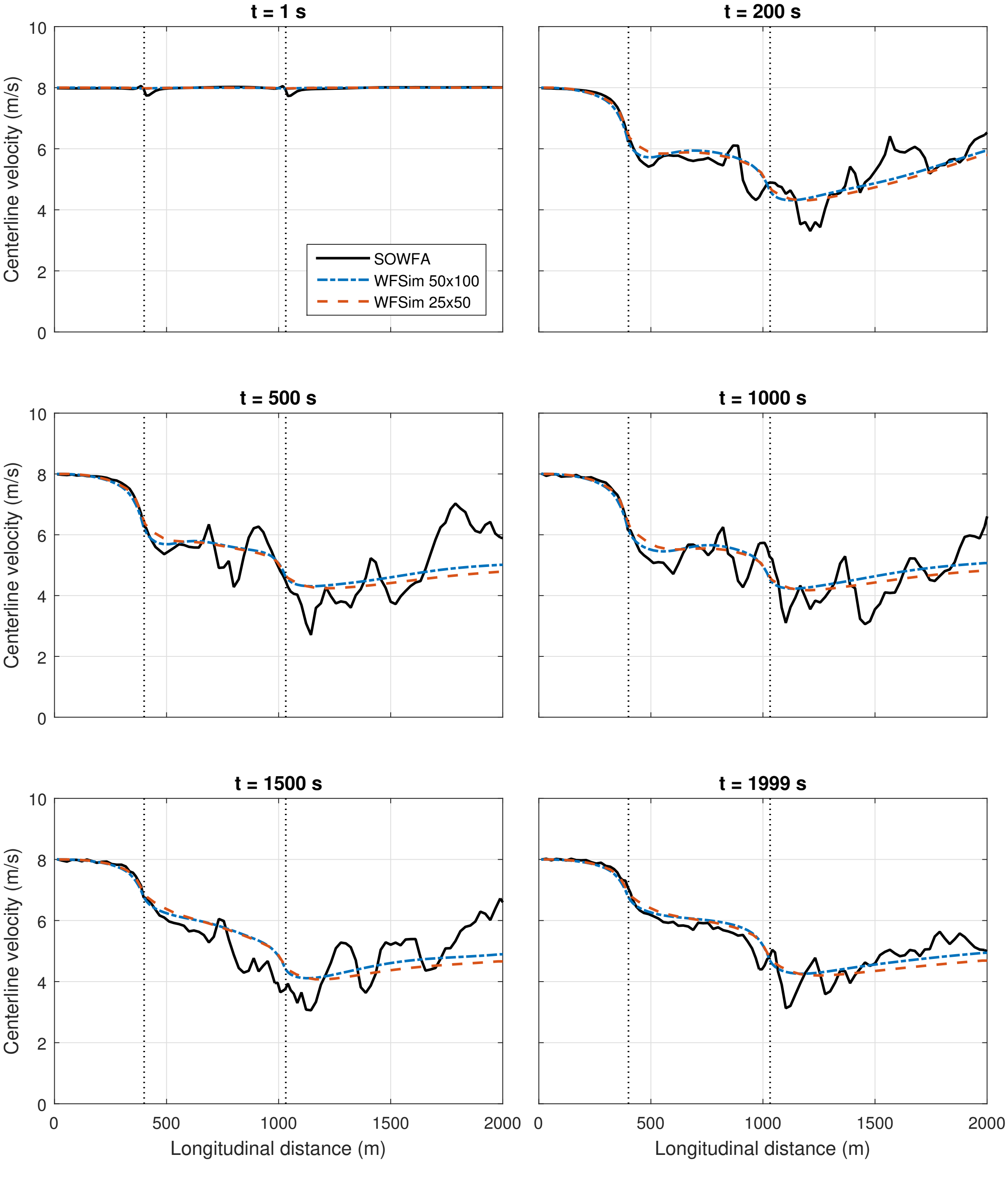


**Figure 2-13:** Mean wake centerlines (m/s) of the SOWFA data, WFSim at a 50x100 meshing, and WFSim at a 25x50 meshing at time instants $t = 1, 200, 500, 1000, 1500, 1999$ s, in absence of an observer. The dotted vertical lines represent the locations of turbine 1 ($x = 400$ m) and turbine 2 ($x = 1032$ m).

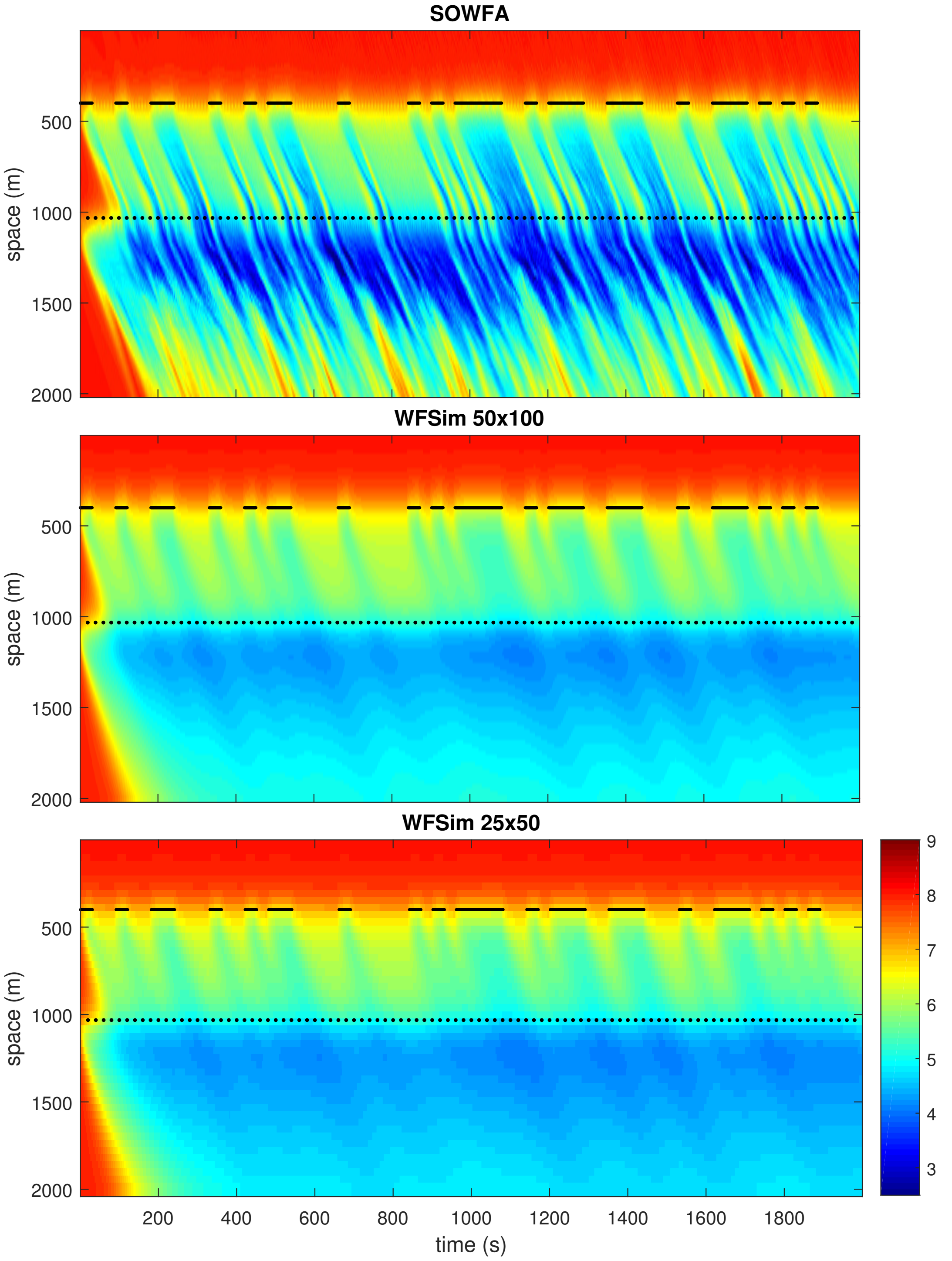


**Figure 2-14:** Mean wake centerline wind speed (m/s) for the raw SOWFA data, and the resulting WFSim simulations at the two meshes. Vertical segments represent the mean wake centerline at a certain time instant. The irregularly black dashed line shows the switching signal of the pitch angle. For the gaps $\beta_w = 0.153$ and for the black lines $\beta_w = 0.278$. Similarly, the regularly dotted line shows the location of turbine 2.

## Chapter 3

# Observer design

The reconstruction of states in a dynamic system is an invaluable part of many modern technological systems. For example, state estimation and prediction is the main focus in weather models, and plays a crucial role in robotics and automotive applications. An introduction to state reconstruction is found in the book by Verhaegen [43]. As mentioned in Section 2-1-11, the crux in observer design for WindFarmSimulator (WFSim) lies with the model size and its nonlinearity.

In this chapter, two time-efficient Kalman filter (KF) designs will be depicted. In Section 3-1, the model of Chapter 2 is manipulated to allow easier observer implementation. The general approach to time-efficient state observers is outlined in Section 3-2, after which the actual filters are presented. The traditional KF, the Approximate Kalman filter (ApKF), and the Ensemble Kalman filter (EnKF) are presented in Sections 3-3 to 3-5, respectively. Finally, additional options in the observer's numerical implementation are outlined in Section 3-6.

## 3-1 Model structure manipulation

The model described in Chapter 2 comes down to a nonlinear state-space system of the form

$$\begin{aligned} E_w(\alpha_k)\alpha_{k+1} =& A_w\ \alpha_k + B_w(\alpha_k)\beta_{w_k} + S_w(\alpha_k), \\ \begin{bmatrix} u_k \\ v_k \end{bmatrix} =& Q_p\ \alpha_k + B_p. \end{aligned} \tag{3-1}$$

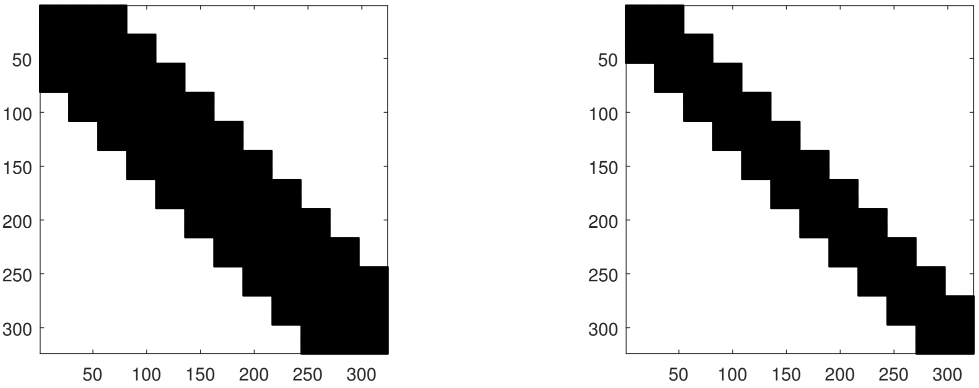
The terms $S_w(\alpha_k)$ and $B_p$ obstruct direct implementation of this model in standard filtering algorithms. To solve this, the system state vector is extended with a constant, known entry of 1. Furthermore, while a number of Kalman filters for implicit, time-varying state-space models exist (e.g., [44, 45]), these algorithms often lead to numerical instability, are too computationally heavy for real-time control with WFSim, and are therefore not further considered in this report. Hence, the implicit system also has to be transformed into an explicit state-space system, resulting in

$$\begin{aligned}\overbrace{\begin{bmatrix}\alpha_{k+1}\\ 1\end{bmatrix}}^{\tilde{\alpha}_{k+1}} &= \overbrace{\begin{bmatrix}E_w(\alpha_k)^{-1}A_w & E_w(\alpha_k)^{-1}S_w(\alpha_k)\\ 0 & 1\end{bmatrix}}^{\tilde{F}(\alpha_k)} \begin{bmatrix}\alpha_k\\ 1\end{bmatrix} + \overbrace{\begin{bmatrix}E_w(\alpha_k)^{-1}B_w(\alpha_k)\\ 0\end{bmatrix}}^{\tilde{B}(\alpha_k)} \beta_{w_k},\\ \begin{bmatrix}\overline{u}_k\\ \overline{v}_k\end{bmatrix} &= \underbrace{\begin{bmatrix}Q_p & B_p\end{bmatrix}}_{\tilde{C}} \begin{bmatrix}\alpha_k\\ 1\end{bmatrix}.\end{aligned} \tag{3-2}$$

$\tilde{F}(\alpha_k)$ in Equation (3-2) is calculated using MATLAB's internal tool for solving linear equalities, `mldivide`. Using `mldivide`, calculating $\tilde{F}(\alpha_k)$ at each time instant is still too computationally costly for practical implementations due to high dimensionality ($N$ is typically of the order $10^3$). Fortunately, the Reverse Cuthill-McKee (RCM) algorithm further reduces computation time by reorganizing the system rows and columns, typically resulting in a reduction in computational effort of 90% or more in WFSim.

**The Reverse Cuthill-McKee algorithm** The Cuthill-Mckee algorithm was first published in a 1969 conference on computing machinery [46], aimed to rearrange sparse matrices into a banded diagonal structure. George and Liu in 1981 improved this algorithm by reversing the index numbers, resulting in the Reverse Cuthill-McKee (RCM) algorithm [47].

Recall that the original matrix structure of $E_w(\alpha_k)$ and $A_w$ have a diagonal-line structure as shown in Figure 2-3. The RCM algorithm reformats these matrices into a sparse banded structure with low bandwidth, as shown in Figure 3-1.



**(a)** Matrix $E_w(\alpha_k) \in \mathbb{R}^{324\times324}$ **(b)** Matrix $A_w \in \mathbb{R}^{324\times324}$

**Figure 3-1:** Location of nonzero elements (structure) in the system matrices for the WFSim farm model in qLPV form under the Reverse Cuthill-McKee (RCM) transformation.

As the RCM algorithm solely rearranges the matrices, no accuracy is lost through this transformation. Furthermore, since the structure of the system matrices does not change throughout time, the correct rearrangement only has to be calculated once, and then applied in each iteration. On a 2012 ultrabook PC, calculating $\tilde{F}(\alpha_k)$ in one specific case takes 130 seconds without the RCM algorithm, and only 4 seconds with the RCM algorithm.[1]

[1] This is for a system with $N = 4559$ states, corresponding to a 50x100 meshing. While this value depends on the PC specifications and simulation settings, it gives an impression of the increase in computational efficiency.

## 3-2 Methodology

For the system of Equation (3-2), standard state observers such as the Kalman filter can readily be implemented. Unfortunately, with increasing system size, the computational cost for these algorithms scale exponentially. For example, in a typically large system with $N = 5 \cdot 10^3$ states (i.e. comparable to WFSim 50x100), a regular Kalman filter typically takes about $8 \cdot 10^2 - 1 \cdot 10^3$ seconds per iteration on a modern ultrabook PC, being several orders of magnitude too slow for real-time applications. To reduce computational costs, one may either scale down the system size, adapt the filtering algorithm, or a combination of both. For both methods, a corresponding filtering algorithm is implemented using WFSim to allow a side-by-side comparison.

**System size reduction** Scaling down a system based on a set of Navier-Stokes (NS) equations is demonstrated in an article by Poussot-Vassal and Sipp [48]. In this paper, the authors are dealing with a high order implicit state-space system in which the system matrices depend on an external parameter: the Reynolds number. The system has the form

$$\begin{aligned} \hat{E}(Re)\dot{\hat{x}}(t) &= \hat{A}(Re)\hat{x}(t) + \hat{B}(Re)\hat{u}(t), \\ \hat{y}(t) &= \hat{C}(Re)\hat{x}(t), \end{aligned}$$

with $\hat{E}$, $\hat{A}$, $\hat{B}$, and $\hat{C}$ system matrices dependent on the Reynolds number $Re$. $\hat{x}(t)$, $\hat{y}(t)$ and $\hat{u}(t)$ are the state vector, system output, and system input at time $t$, respectively. The system is reduced according to three steps. Firstly, the original NS equations are discretized spatially and parametrically for a number of Reynolds numbers. Secondly, for each Reynolds number a very large linear time-invariant (LTI) system ($7 \cdot 10^5$ states) is obtained, which is reduced in size to 18 states by using the Iterative Tangential Interpolation Algorithm (IRKA). Thirdly, the linearized low-order models at different Reynolds numbers are interpolated and transformed into a low-complexity linear fractional representation.

The problem with system reduction following Poussot-Vassal and Sipp's method is that the system matrices of WFSim do not depend on an external parameter such as the Reynolds number, but rather depend on the internal states. Linearization of WFSim is ambiguous, and would defeat the purpose of having such a sophisticated medium fidelity model as WFSim, as many linear flow models are readily available. The power in this model lies in its format (state-space model), while maintaining high accuracy through nonlinearity and a relatively large number of states. This allows WFSim to be applicable to different wind directions and wind speeds, and allows the inclusion of both axial-induction-based control as wake redirection based control. Linearizing the model with respect to each parameter would yield a huge lookup-table, rendering this solution infeasible.

An alternative to linearization would be to simplify the model by neglecting certain matrix entries in $\tilde{F}$, $\tilde{B}$ or $\tilde{C}$. In this work, the ApKF to be presented in Section 3-4 relies on a simplification of the system matrices in combination with a traditional KF. Rather than reducing the system size, the system matrices are sparsified in the algorithm to reduce the number of floating point operations per iteration.

**Time-efficient state observers** To reduce computational cost in filtering updates, one may opt for time-efficient algorithms while leaving the flow model intact. Since WFSim is nonlinear, a KF is not necessarily an optimal filter as it would be for an LTI system. Furthermore, Kalman filtering for large order systems is a common problem often tackled by specific KF variants such as the EnKF, performing much better in computational cost and often too in accuracy for nonlinear dynamics compared to the traditional KF [49]. In this work, the EnKF to be presented in Section 3-5 relies on a time-efficient variant of the traditional KF that leaves the system matrices untouched.

## 3-3 Traditional Kalman filtering

Before the time-efficient filter variants are discussed, first the traditional Kalman filter (KF) is elaborated on. The KF is an observer that is designed to find the statistically optimal filtered state estimate $\hat{\tilde{\alpha}}_{k|k}$ and predicted estimate $\hat{\tilde{\alpha}}_{k+1|k}$ recursively for a system with white measurement and process noise. Statistically optimal is typically defined as unbiased, $E\left[e_k\right] = 0$ where $e_k$ is the state estimation error, and with minimum expected state estimation error variance, $P_k = E\left[e_k e_k^T\right]$. In both equations, $E$ denotes *expectation*, as in probability theory [43]. The update algorithm for a system of the form of Equation (3-2) is shown in Figure 3-2.

1. Filtered state estimate
   (a) $P_{k|k} = P_{k|k-1} - P_{k|k-1}\tilde{C}^T\left(R_k + \tilde{C}P_{k|k-1}\tilde{C}^T\right)^{-1}\tilde{C}P_{k|k-1}$,
   (b) $L'_k = P_{k|k-1}\tilde{C}^T\left(R_k + \tilde{C}P_{k|k-1}\tilde{C}^T\right)^{-1}$,
   (c) $\hat{\tilde{\alpha}}_{k|k} = \tilde{\alpha}_{k|k-1} + L'_k\left(\begin{bmatrix}\overline{u}_k \\ \overline{v}_k\end{bmatrix}_{\text{measured}} - \tilde{C}\hat{\tilde{\alpha}}_{k|k-1}\right)$,

2. Predicted state estimate
   (a) $P_{k+1|k} = \tilde{F}_k P_{k|k-1}\tilde{F}_k^T + Q_k - \left(S_k + \tilde{F}_k P_{k|k-1}\tilde{C}^T\right)\left(R_k + \tilde{C}P_{k|k-1}\tilde{C}^T\right)^{-1}\ldots$
   $\times\left(S_k + \tilde{F}_k P_{k|k-1}\tilde{C}^T\right)^T$,
   (b) $L_k = (S_k + \tilde{F}_k P_{k|k-1}\tilde{C}^T)(R_k + \tilde{C}P_{k|k-1}\tilde{C}^T)^{-1}$,
   (c) $\hat{\tilde{\alpha}}_{k+1|k} = \tilde{F}_k\hat{\tilde{\alpha}}_{k|k-1} + \tilde{B}_k\beta_{w_k} + L_k\left(\begin{bmatrix}\overline{u}_k \\ \overline{v}_k\end{bmatrix}_{\text{measured}} - \tilde{C}\hat{\tilde{\alpha}}_{k|k-1}\right)$,

**Figure 3-2:** Filtering algorithm for the KF. New measurements are fed into the system at time $k$, upon which the KF provides an optimal estimate for $\tilde{\alpha}_k$.

with $P_{k|k-1} = E\left[(x_k - \hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1})^T\right]$ the expected state estimation error covariance matrix at time $k$ with information up until time $k-1$. $\tilde{F}_k$ and $\tilde{B}_k$ are a compact notation of the system matrices $\tilde{F}(\alpha_k)$ and $\tilde{B}(\alpha_k)$. $L_k$ is the Kalman observer gain at time $k$. Covariances $R_k$, $Q_k$ and $S_k$ are defined as

$$E\left[\begin{bmatrix}\tilde{v}_i \\ \tilde{w}_i\end{bmatrix}\begin{bmatrix}\tilde{v}_j^T & \tilde{w}_j^T\end{bmatrix}\right] = \begin{bmatrix}R_k & S_k^T \\ S_k & Q_k\end{bmatrix}\Delta_{i-j}, \tag{3-3}$$

with $\tilde{v}_k$ and $\tilde{w}_k$ the measurement and process noise at time $k$, respectively, and $\Delta_{i-j}$ the unit pulse being nonzero only for $i = j$ [43]. While these covariance matrices should correspond to the true noise in a physical system (e.g., in measurement equipment), it is often regarded as a tuning parameter due to limited availability of information about the true system. For an LTI system, the KF converges to a steady-state optimal gain $L^\star$,

$$\lim_{k\to\infty} L_k = L^\star = \left(S_{\text{LTI}} + A_s P^\star C_s^T\right)\left(C_s P^\star C_s^T + R_{\text{LTI}}\right)^{-1}, \tag{3-4}$$

with $P^\star$ a solution to the discrete algebraic Riccati equation

$$P^\star = A_s P^\star A_s^T + Q_{\text{LTI}} - (S_{\text{LTI}} + A_s P^\star C_s^T)(C_s P^\star C_s^T + R_{\text{LTI}})^{-1}(S_{\text{LTI}} + A_s P^\star C_s^T)^T. \tag{3-5}$$

In the situation that $(A_s, C_s)$ is observable[2] and $(A_s, Q_{\text{LTI}}^{1/2})$ is reachable[3], $A_s - L^\star C_s$ is asymptotically stable. The steady-state Kalman filter is typically implemented, as the improvements with the time-varying filter typically do not weigh up against the additional implementation and computational efforts.

The standard KF takes in the order of $10^3$ seconds per iteration for a system with $N = 4559$ states (i.e., comparable to WFSim 50x100) on an ultrabook PC, which is much too high for real-time implementations. This value depends on the PC specifications and simulation settings, but it gives an impression of the computational inefficiency of this algorithm.

For more information on the standard KF, please see the book by Verhaegen [43].

## 3-4 Approximate Kalman filtering

To save on computational cost in the traditional KF outlined in Section 3-3, the algorithm described in Figure 3-2 is analyzed and the most resourceful operations are adapted, namely the update of the covariance matrix $P_{k+1|k}$:

$$\begin{aligned} P_{k+1|k} = \tilde{F}_k P_{k|k-1} \tilde{F}_k^T + Q_k - \left(S_k + \tilde{F}_k P_{k|k-1} \tilde{C}_k^T\right)\left(R_k + \tilde{C}_k P_{k|k-1} \tilde{C}_k^T\right)^{-1} \ldots \\ \times \left(S_k + \tilde{F}_k P_{k|k-1} \tilde{C}_k^T\right)^T \end{aligned}$$

The most heavy operation is the multiplication of the three square matrices $\tilde{F}_k P_{k|k-1} \tilde{F}_k^T$, requiring $2N^3$ floating point operations.[4] In the case of 4559 states, this results in $1.90 \cdot 10^{11}$ operations, yielding a high computational time ($10^2 - 10^3$ seconds). Hence, both matrices present in this equation, $P_{k|k-1}$ and $\tilde{F}_k$, are modified.

1. The state covariance matrices $P_k$ are sparsified by neglecting all off-diagonal elements. This is justified by noting that the off-diagonal elements typically are of order $10^1 - 10^2$

[2] A system is observable if the system outputs can be mapped back to the system state vector uniquely over a finite time interval.

[3] A system is reachable if it can be controlled to any desired final state based on a certain sequence of control inputs, in any finite time.

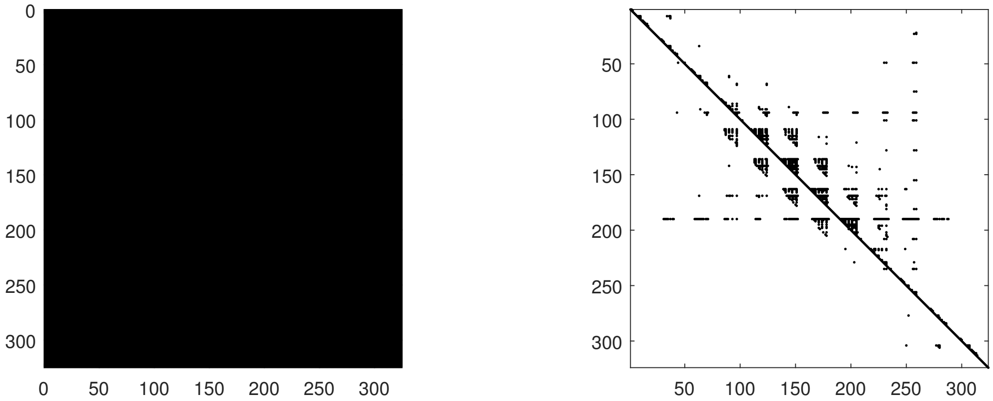
[4] In WFSim, the calculation of $E^{-1}A$ is a computationally intensive computation too, but besides applying the RCM algorithm it is ambiguous how to further reduce computational cost in this operation.

smaller than the diagonal elements. The implicit assumption underlying this sparsification is that the state vector $\alpha_k$ entries are uncorrelated. Based on physical intuition, it is hard to tell the validity of this assumption, since the entries of $\alpha_k$ do not have a direct physical interpretation due to the projection of Equation (2-25). This simplification reduces the computational cost from $2N^3$ to approximately $N^3$ for large $N$, decreasing the amount of operations by almost 50% for certain simulation cases.

2. Furthermore, the system matrix $\tilde{F}(\alpha_k)$ is sparsified by neglecting all entries $[i, j]$ that meet the condition

$$|\tilde{F}(\alpha_0)[i,j] \cdot \alpha_0(j)| < \overline{(|\tilde{F}(\alpha_0)[i,:] \cdot \alpha_0|)} \cdot z, \tag{3-6}$$

with $z$ typically around $0.05 - 0.10$, and $i$ and $j$ denoting the row and column of the matrix, respectively. For $z = 0$, the original $\tilde{F}$ matrix is retrieved. The parameter $z$ is tuned as a trade-off between computational efficiency and model accuracy. This sparsification typically renders over 95% of the system entries sparse for large $N$, severely cutting down on computational cost. Sparsification of $\tilde{F}(\alpha_k)$ is demonstrated in Figure 3-3.



**(a)** Before sparsification.

**(b)** After sparsification with $z = 0.05$, as used in the ApKF. Only 1.3% nonzero elements remain.

**Figure 3-3:** Location of nonzero elements (matrix structure) of $\tilde{F}(\alpha_k) \in \mathbb{R}^{324\times324}$.

In simulation, the location of nonzero elements is stored in memory, and all other elements are set to zero at each time instant. Hence, this costly computation is only to be performed once prior to simulation.

The sparsification is justified by choosing $\alpha_0$ to correspond to a fully developed flow. Then, as also previously seen in Figure 2-10, the flow field does not change enough to significantly modify the location of dominant entries in $\tilde{F}(\alpha_k)$. Note that currently only axial-induction-based control is assessed in simulation, and this sparsification may be invalid for wake redirection based control. Furthermore, it is assumed that the wind speed and inflow direction are constant. Changes in wind direction are expected to change the degree of interdependency between states, thereby changing the location of dominant matrix entries in $\tilde{F}(\alpha_k)$.

Furthermore $S_k$ is neglected to reduce the number of tuning parameters, implying there is no cross-correlation between measurement and process noise.

These simplifications, while significantly speeding up the filtering algorithm by decreasing the computational effort from $2N^3$ to $O(N) - O(N^2)$, do not noticeably impact the accuracy of the filter, as demonstrated in Figure 3-4.

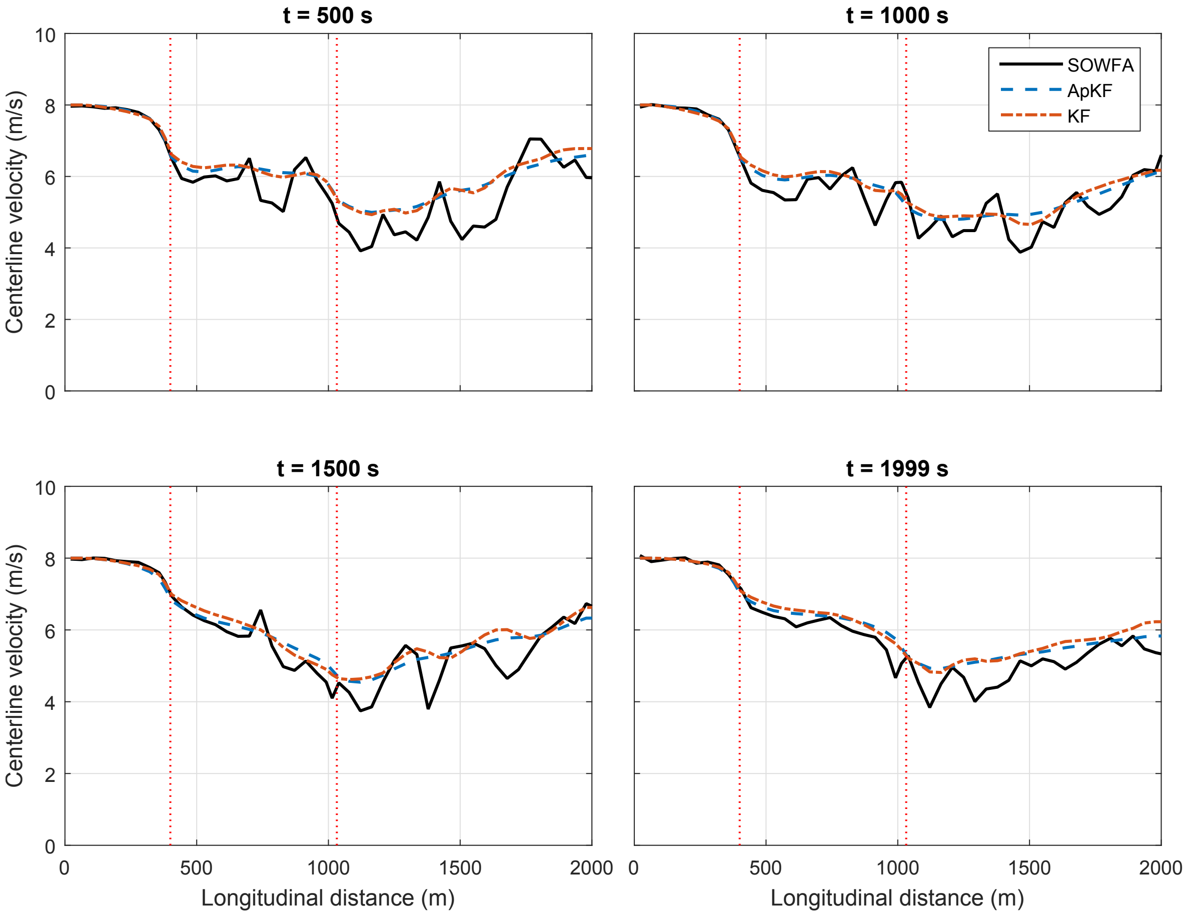


**Figure 3-4:** Mean wake centerlines: a comparison of the ApKF ($z = 0.05$) with the full KF at 25x50 meshing. While being equally accurate, the ApKF is a factor $10^1$ faster at this meshing resolution (25x50), and a factor $10^2$ at a 50x100 meshing, as will be depicted in Chapter 4 (along with simulation settings).

This figure shows the true (SOWFA) and predicted mean wake centerlines at times $t = 500, 1000, 1500, 2000$ s at the 25x50 meshing. Simulation settings and results will be the topic of Chapter 4. The main purpose of here is to demonstrate the similarity in state estimation performance between the ApKF and the KF, while yielding a reduction in iteration time of a factor $10^1 - 10^2$. Benefits of the ApKF scale up with model size, as will be demonstrated in Chapter 4.

Finally, note that in both the regular KF and this ApKF, the last covariance matrix entry is 0. This is because the qLPV system was extended by a constant *known* entry 1. Thus, the covariance matrices have the form

$$P'_0 = \begin{bmatrix} P_0 & 0 \\ 0 & 0 \end{bmatrix}, \quad (3\text{-}7) \qquad P'_k = \begin{bmatrix} P_k & 0 \\ 0 & 0 \end{bmatrix}, \quad (3\text{-}8) \qquad Q'_k = \begin{bmatrix} Q_k & 0 \\ 0 & 0 \end{bmatrix}. \quad (3\text{-}9)$$

## 3-5 Ensemble Kalman filtering

The second time-efficient filter implemented is the Ensemble Kalman filter (EnKF), belonging to the group of particle filtering methods [50]. The EnKF is a suboptimal variant of the KF that replaces the covariance matrix with a sample covariance in order to reduce computational cost and improve performance in nonlinear systems. The EnKF has been used in atmospheric applications (e.g., [51, 52, 53, 54]) and more rarely in other applications such as traffic estimation (e.g., [55]) and target tracking (e.g., [56]). A more exhaustive list of known applications in literature is described in a paper by Evensen [49].

An introduction to Ensemble Kalman filtering is given in an article by Houtekamer [54]. Furthermore, this paper demonstrates an implementation at the Canadian Meteorological Centre using only a small number of ensembles of order $O(10^2)$ for a system with several orders of magnitude more states. The main motivations for using an EnKF implementation are:

1. It deals well with high order systems of $10^3 - 10^6$ states and similarly a large number of observations. Typically, the EnKF is applied in weather forecasting where such models are used. Note that, even for these large systems, an ensemble size of several orders of magnitude smaller typically still yields good performance [50, 54].
2. It is well suited for nonlinear systems. The EnKF algorithm propagates the ensemble members and covariance matrices using the full nonlinear model, thereby properly dealing with saturation of errors [54]. Unlike the (Extended) KF, no linearized equations of the respective model are required.
3. The algorithm is easy to parallelize due to identical operations performed for each ensemble member at each iteration.
4. It is computationally efficient due to the ensemble size typically being of several orders of magnitude smaller than the number of model states.

The algorithm used in this work is largely based on an article by Evensen [49]. This paper introduces the reader both to the theoretical and practical aspects of the EnKF. The algorithm described in this paper has been given the benefit of implementation due to including an exhaustive instruction for numerical implementation, aimed at computational efficiency and numerical stability.

First, the filtering algorithm will be described in Section 3-5-1. Secondly, the initial ensemble distribution will be depicted in Section 3-5-2. Finally, a few remarks shall be made on the numerical implementation in Section 3-5-3.

### 3-5-1 Filtering algorithm

The EnKF is a sequential data assimilation method, just like the traditional KF. First, we introduce a forecasted state vector $\psi^f_{j,k}$ as an estimate to $\tilde{\alpha}_k$ with information up until time $k-1$, and filtered state vector $\psi^a_{j,k}$ as an estimate to $\tilde{\alpha}_k$ with information up until time $k$. $1 \leq j \leq n_e$ is an integer value used for numbering. Secondly, we introduce the forecasted ensemble $A^f_{e_k} \in \mathbb{R}^{N \times n_e}$, the filtered ensemble $A^a_{e_k} \in \mathbb{R}^{N \times n_e}$ and their means $\overline{A^f_{e_k}} \in \mathbb{R}^{N \times n_e}$

and $\overline{A^a_{e_k}} \in \mathbb{R}^{N \times n_e}$ as

$$A^f_{e_k} = \begin{bmatrix} \psi^f_{1,k} & \psi^f_{2,k} & \cdots & \psi^f_{n_e,k} \end{bmatrix}, \tag{3-10}$$

$$A^a_{e_k} = \begin{bmatrix} \psi^a_{1,k} & \psi^a_{2,k} & \cdots & \psi^a_{n_e,k} \end{bmatrix}, \tag{3-11}$$

$$\overline{A^f_{e_k}} = A^f_{e_k} \cdot 1_{n_e}, \tag{3-12}$$

$$\overline{A^a_{e_k}} = A^a_{e_k} \cdot 1_{n_e}, \tag{3-13}$$

with $n_e$ the number of ensemble members, and $1_{n_e} \in \mathbb{R}^{n_e \times n_e}$ the matrix with each element equal to $n_e^{-1}$. Thus, each column in an ensemble $A_{e_k}$ is an estimate to $\tilde{\alpha}_k$. All columns in a mean ensemble $\overline{A_{e_k}}$ are identical and represent one averaged estimated state vector for $\tilde{\alpha}_k$, obtained by taking the mean of all the columns in $A_{e_k}$. This average of all $n_e$ hypothesized state vectors is defined as the optimal estimate out of the ensemble for time $k$, namely $\psi^a_{\text{opt},k}$ for the filtered optimal state estimate or $\psi^f_{\text{opt},k}$ for the predicted optimal state estimate.

Furthermore, the respective covariance matrices for model forecast, analysis and measurements at time $k$ are denoted by $P^f_k$, $P^a_k$ and $R_k$, respectively (i.e., $P_{k|k-1} = P^f_k$ and $P^a_k = P_{k|k}$). Representing this using a finite sample covariance, this turns into $P^f_{e_k} \approx P^f_k$, $P^a_{e_k} \approx P^a_k$ and $R_{e_k} \approx R_k$.[5] The sample covariance matrices are calculated from the ensemble $A_{e_k}$ according to

$$P^f_{e_k} = \frac{(A^f_{e_k} - \overline{A^f_{e_k}})(A^f_{e_k} - \overline{A^f_{e_k}})^T}{n_e - 1}, \tag{3-14}$$

$$P^a_{e_k} = \frac{(A^a_{e_k} - \overline{A^a_{e_k}})(A^a_{e_k} - \overline{A^a_{e_k}})^T}{n_e - 1}, \tag{3-15}$$

$$R_{e_k} = \frac{\gamma_{e_k} \gamma^T_{e_k}}{n_e - 1}, \tag{3-16}$$

in which $\gamma_{e_k}$ is the matrix with (artificial) measurement disturbances at time $k$, as will be defined later this paragraph. The state sample covariance matrix in the limit $n_e \to \infty$ will come down to the full covariance matrix $P^a_k$ used in the traditional KF.

When measurements at time $k$ are fed into the observer, the analysis algorithm to determine the optimal filtered estimate for $\tilde{\alpha}_k$ is shown in Figure 3-5.

## 3-5-2 Initializing the ensemble

Ideally, the initial ensemble $A^a_{\text{e}0}$ should represent the initial state covariance matrix $P_0$ by Equation (3-15). The ensemble is initialized as a set of hypothesized state vectors, typically of about $10^2$ ensemble members. Assuming no knowledge of the initial flow field, in WFSim the members are distributed equidistantly as different uniform flow fields, each with a different freestream longitudinal and lateral velocity. The fields are generated according to the number of ensemble members and a predefined range of velocities.

[5] In sample covariance matrices, covariance matrices are approximated using a finite set of data. In the limit of an infinite amount of data, the sample covariance is equal to the true covariance.

1. The set of measurements at time $k$ is collected into a vector $d_k$, duplicated $n_e$ times and summed with artificial noise $\epsilon^m$ to form $d_{1,k} \dots d_{n_e,k}$. These vectors are collected in a single matrix $D_{e_k}$, resulting in

$$d_{j,k} = d_k + \epsilon^m_{j,k}, \tag{3-17}$$

$$\gamma_{e_k} = \begin{bmatrix} \epsilon^m_{1,k} & \epsilon^m_{2,k} & \cdots & \epsilon^m_{n_e,k} \end{bmatrix}, \tag{3-18}$$

$$D_{e_k} = \begin{bmatrix} d_{1,k} & d_{2,k} & \cdots & d_{n_e,k} \end{bmatrix}, \tag{3-19}$$

   with $\epsilon^m_{j,k}$ a vector of Gaussian measurement noise with standard deviation $\sigma_{\epsilon^m}$. The goal is to have this artificial noise identical to the real noise present in the measurements. Then, the sample covariance matrix of the measurements is defined as $R_e = \overline{\epsilon^m \epsilon^{mT}}$. In the limit $n_e \to \infty$, then $R_e \to R_k$.

2. Each ensemble member (thus: each column of $A^a_{e_{k-1}}$) is updated forward in time to $A^f_{e_k}$ according to a slightly modified version of Equation (3-2), namely

$$\begin{aligned} \psi^f_{j_k} &\approx \tilde{F}(\psi^a_{\text{opt},k-1})\psi^a_{j,k-1} + \tilde{B}(\psi^a_{\text{opt},k-1})\beta_{w_{k-1}} + \epsilon^p_j \\ \begin{bmatrix} \overline{u}_k \\ \overline{v}_k \end{bmatrix}^f &= \tilde{C}\psi^f_{j,k}, \end{aligned} \tag{3-20}$$

   with $\epsilon^p$ process noise, and $\psi^a_{\text{opt},k}$ the optimal state estimate defined as the mean of all ensemble members $\overline{A^a_{k-1}}$. The process noise covariance matrix is defined as $Q_e = \overline{\epsilon^p(\epsilon^p)^T}$, in which $\epsilon^p$ is supposed to represent the true process noise in the system. This process is highly parallelizable. All forecasted state vectors are collected in $A^f_{e_k}$ according to Equation (3-10).

   Note that there is a modification from Evensen's original algorithm in Equation (3-20). Namely, $\tilde{F}$ and $\tilde{B}$ only have to be calculated once per time instant in Equation (3-20), rather than $n_e$ times in Evensen's algorithm. Also, note that cross-covariance $S_k$ is neglected in this filter.

3. Each ensemble member is updated according to the analysis equation

$$A^a_{e_k} = A^f_{e_k} + P^f_{e_k}\tilde{C}^T \left(\tilde{C}P^f_{e_k}\tilde{C}^T + R_{e_k}\right)^{-1} \left(D_{e_k} - \tilde{C}A^f_{e_k}\right), \tag{3-21}$$

   containing all updated ensemble members at time instant $k$ using measurements up until time $k$: $\psi^a_{j,k}$ for $j = 1 \dots n_e$. Note that the inversion in Equation (3-21) has potential singularities, which is the main motivation for using the pseudo-inverse as will be explained in Section 3-5-3.

4. The system is now propagated one time instant further, new measurements are obtained, and the process is repeated starting with step 1 of the algorithm.

**Figure 3-5:** Filtering algorithm for the EnKF. New measurements are fed into the system at time $k$, upon which the EnKF updates all ensemble members and provides an optimal estimate for $\tilde{\alpha}_k$.

### 3-5-3 Numerical implementation

Evensen [49] describes a detailed numerical implementation for the EnKF aimed at numerical stability and computational efficiency. Specifically, it optimizes the order of matrix operations to minimize the number of floating point operations, and thereby reduce computational cost. First, Equation (3-21) is modified by defining

$$A'_{e_k} = A^f_{e_k} - \overline{A^f_{e_k}}, \tag{3-22}$$

$$D'_{e_k} = D_{e_k} - \tilde{C}A^f_{e_k}, \tag{3-23}$$

yielding the modified analysis update

$$A^a_{e_k} = A^f_{e_k} + A'_{e_k}A'^T_{e_k}\tilde{C}^T\left(\tilde{C}A'_{e_k}A'^T_{e_k}\tilde{C}^T + \gamma_{e_k}\gamma^T_{e_k}\right)^{-1} D'_{e_k}. \tag{3-24}$$

The inverse is calculated using a singular value decomposition (SVD). In the case that measurement perturbations $\gamma_{e_k}$ are chosen such that $\tilde{C}A'_{e_k}\gamma^T_{e_k} \equiv 0$, the following holds[6]

$$\tilde{C}A'_{e_k}A'^T_{e_k}\tilde{C}^T + \gamma_{e_k}\gamma^T_{e_k} = (\tilde{C}A'_{e_k} + \gamma_{e_k})(\tilde{C}A'_{e_k} + \gamma_{e_k})^T. \tag{3-25}$$

Defining a SVD as $\tilde{C}A'_{e_k} + \gamma_{e_k} = U_{\text{SVD}}\Sigma V^T_{\text{SVD}}$, the following is obtained

$$\tilde{C}A'_{e_k}A'^T_{e_k}\tilde{C}^T + \gamma_{e_k}\gamma^T_{e_k} = U_{\text{SVD}}\Sigma V^T_{\text{SVD}}V_{\text{SVD}}\Sigma^T U^T_{\text{SVD}} = U_{\text{SVD}}\Sigma\Sigma^T U^T_{\text{SVD}} = U_{\text{SVD}}\Lambda U^T_{\text{SVD}}. \tag{3-26}$$

From this SVD the least dominant eigenvalues ($< 0.1\%$) are neglected to speed up computations. The numerical update algorithm is outlined in Figure 3-6.

1. The entire ensemble is updated forward in time according to

   $$A^f_{e_k} = \tilde{F}(\psi^f_{\text{opt},k})A^a_{e_k} + \left[\tilde{B}(\psi^f_{\text{opt},k})\beta_{w_k} + \epsilon^p_1 \quad \tilde{B}(\psi^f_{\text{opt},k})\beta_{w_k} + \epsilon^p_2 \quad \cdots \quad \tilde{B}(\psi^f_{\text{opt},k})\beta_{w_k} + \epsilon^p_{n_e}\right]$$

2. The SVD of $\tilde{C}A'_{e_k} + \gamma_e$ is calculated, yielding $U_{\text{SVD}}$ and $\Sigma$. Then, $\Lambda = \Sigma\Sigma^T$.
3. $\Lambda$ is made sparse by neglecting all eigenvalues with small contribution ($< 0.1\%$).
4. Computational efficiency is achieved by arranging the order of operations:
   i. Calculate $X_1 = \Lambda^{-1}U^T_{\text{SVD}}$.
   ii. Calculate $X_2 = X_1(D_{e_k} - \tilde{C}A^f_{e_k})$.
   iii. Calculate $X_3 = U_{\text{SVD}}X_2$.
   iv. In the case of a large number of observations: $A^a_{e_k} = A^f_{e_k} + A'_{e_k}\left((\tilde{C}A'_{e_k})^T X_3\right)$.
   In the case of a small number of observations: $A^a_{e_k} = A^f_{e_k} + \left(A'_{e_k}(\tilde{C}A'_{e_k})^T\right)X_3$.

**Figure 3-6:** Suggested numerical implementation of the EnKF.

[6] This implies that the ensemble perturbations are uncorrelated.

## 3-6 Implementation in WFSim

The three filters described in this chapter, respectively the KF, ApKF and EnKF, are each implemented using the WFSim farm model.[7]

The observers rely on the amount and location of wind speed measurements for their analysis updates. Naturally, more measurements results in better observer performance. Currently, measurements around the turbines and at equidistant locations throughout the grid are incorporated. Work is currently ongoing to optimize the location of measurements based on a linearized version of the WFSim model. As this work progresses, better performance is expected with an equal amount of measurements.

Furthermore, for debugging purposes, a number of quantitative and qualitative measures are implemented. While this does slow down iteration time, it is very insightful for observer development and tuning. This includes:

- A figure of the initial mesh and measurement locations.
- A figure containing contour plots of the flow at every $*$ iterations.
- A figure of the wake centerline at every $*$ iterations.
- The root mean square (RMS) error, variance accounted for (VAF) and quality of fit (QOF) between the estimated and true wake centerline in m/s.
- The iteration time for the forecast and analysis updates.
- Maximum and average estimation error in m/s.
- Different levels of written output, for serious debugging to regular usage.

Also, for observer testing, noise can be introduced to the system in a number of ways:

- Noise in the measurements.
- Noise in the initial flow field.
- Noise in the freestream velocity.
- Noise in the input vector $\beta_w$.
- Noise in the initial state vector $\alpha_0$.

Additionally, a number of other options are included:

- A predefined random seed can be loaded to have consistent noise, allowing a one-to-one comparison in performance between different simulations.
- An observer delay can be introduced. Thus, the observer will be initialized after a defined number of time steps.
- Figures, simulation settings, observer settings, and estimated flow fields can be saved optionally to external files for future revision.

The framework is modular, using much of the existing function files of the original WFSim source code. Updates require manual labor, but are relatively straight-forward and should take no longer than a couple of hours (depending on the number of changes in WFSim). New observers can easily be implemented. While no detailed manual exists, the code is self-explanatory and many comments are included to help those unfamiliar with the model.

[7] To get more information and possibly (parts of) the source code, please contact the author at B.M.Doekemeijer@student.tudelft.nl, or the related research group from the Delft University of Technology at S.Boersma@tudelft.nl and J.W.vanWingerden@tudelft.nl.

Chapter 4

# Simulation results and discussion

In this chapter, the simulation results using the WindFarmSimulator (WFSim) wind farm model (previously discussed in Chapter 2) with respectively the Ensemble Kalman filter (EnKF) and the Approximate Kalman filter (ApKF) (previously discussed in Chapter 3) will be displayed and discussed. This chapter clearly shows the contribution of this work to current literature: a first step to higher fidelity dynamic closed-loop control of wind farms.

The simulation scenario is depicted in Section 4-1. Measures used to quantify performance of the observers are outlined in Section 4-2. The way in which the observers are tuned is discussed in Section 4-3. The results for the ApKF and the EnKF are displayed in Sections 4-4 and 4-5, respectively. Finally, an overview and discussion of all results is given in Section 4-6.

## 4-1 Simulation scenario

The goal is to test the observer designs of Chapter 3 for WFSim under realistic conditions with high fidelity data obtained from Simulator fOr Wind Farm Applications (SOWFA). This implies a limited number of observations, noise on the measurements, uncertain input signal $\beta_w$, unknown boundary conditions, and limited computational resources. More precisely, the following settings were chosen:

- The simulation setup described in Section 2-3 is also used for observer testing, including atmospheric and freestream flow conditions.
- 11% (for the 50x100 meshing) or 23% (for the 25x50 meshing) of the measurements are available to the observer for state reconstruction. The exponential mesh including the measurement locations are displayed in Figure 4-1. Note that this is not very realistic, and in real wind farms much less observations will be available for state reconstruction. Here, the goal is to push the limits of the observer.[1]

[1] Note that, in absolute sense, 11% of the measurements in the 50x100 grid is a larger number than 23% in a 25x50 grid.

- Gaussian noise with standard deviation $\sigma = 0.10$ m/s is added to the measurements. This is comparable to the resolution of current commercial lidar systems [57].
- The input signal $\beta_w$ is chosen to be according to Figure 2-9. Recall from Chapter 2 that this estimate, especially for the second turbine, already contains much noise, and thus no additional noise is introduced.
- The observers will be initialized from a homogeneous flow field. For the EnKF, this means $n_e$ homogeneous flow fields, a different one for each ensemble member.
- The observer will be initialized from the first time instant: no artificial delay will be introduced between the to be estimated system dynamics and the observer.
- No other noise is introduced. Model error is already very much present due to SOWFA including the third (vertical) dimension and significantly more sophisticated turbulence and rotor modeling.

## 4-2 Performance measures

To judge the performance of an observer and for tuning purposes, a number of quantitative measures are taken into account. Specifically, for our application, these measures include:

1. The fit between the mean wake centerlines of SOWFA and WFSim.[2] Three different measures are employed for this, namely:
   - The root mean square (RMS) error. For two lines, estimated data $y \in \mathbb{R}^Z$ and true data $z \in \mathbb{R}^Z$, this is calculated according to

     $$\text{RMS}(y, z) = \sqrt{\frac{1}{Z} \cdot [(y_1 - z_1)^2 + (y_2 - z_2)^2 + \cdots + (y_Z - z_Z)^2]}, \tag{4-1}$$

     providing a quantity of the average error between the two datasets.
   - The variance accounted for (VAF) [43]. This gives a measure of similarity between the dynamics in two datasets. VAF is calculated according to

     $$\text{VAF}(y, z) = \left(1 - \frac{\text{variance}(z - y)}{\text{variance}(z)}\right) \cdot 100\%. \tag{4-2}$$

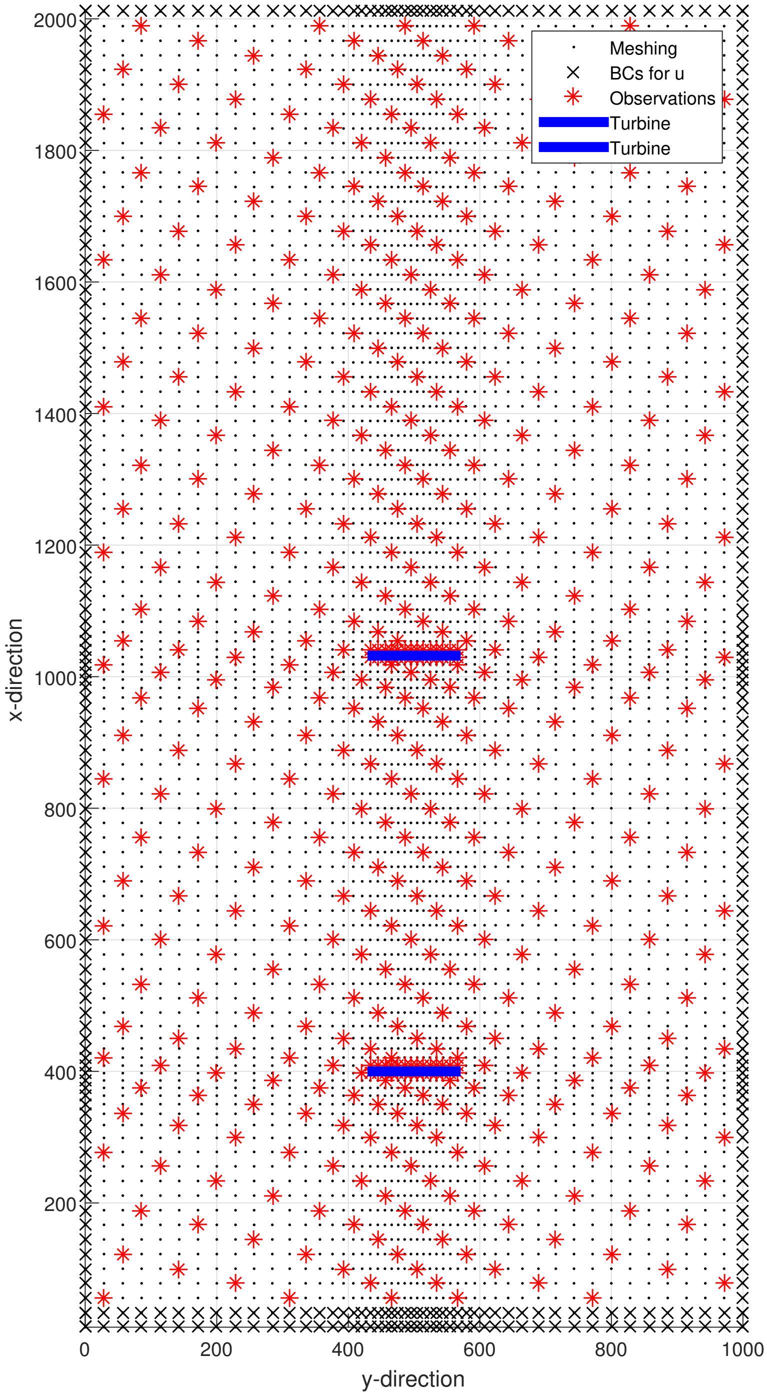
   - The quality of fit (QOF) [43]. This measure both includes the similarity in dynamics of two datasets and their absolute difference. Thus, the QOF is a single measure to get a total picture of the observer's performance.

     $$\text{QOF}(y, z) = \max\left(\left[0, 1 - \frac{||z - y||_2^2}{||z||_2^2}\right]\right) \cdot 100\%. \tag{4-3}$$

   Ideally, one would like to minimize the RMS error and maximize the VAF and QOF.
2. The average computational effort (duration) for one iteration.
3. The maximum and average estimation error throughout the entire flow field.

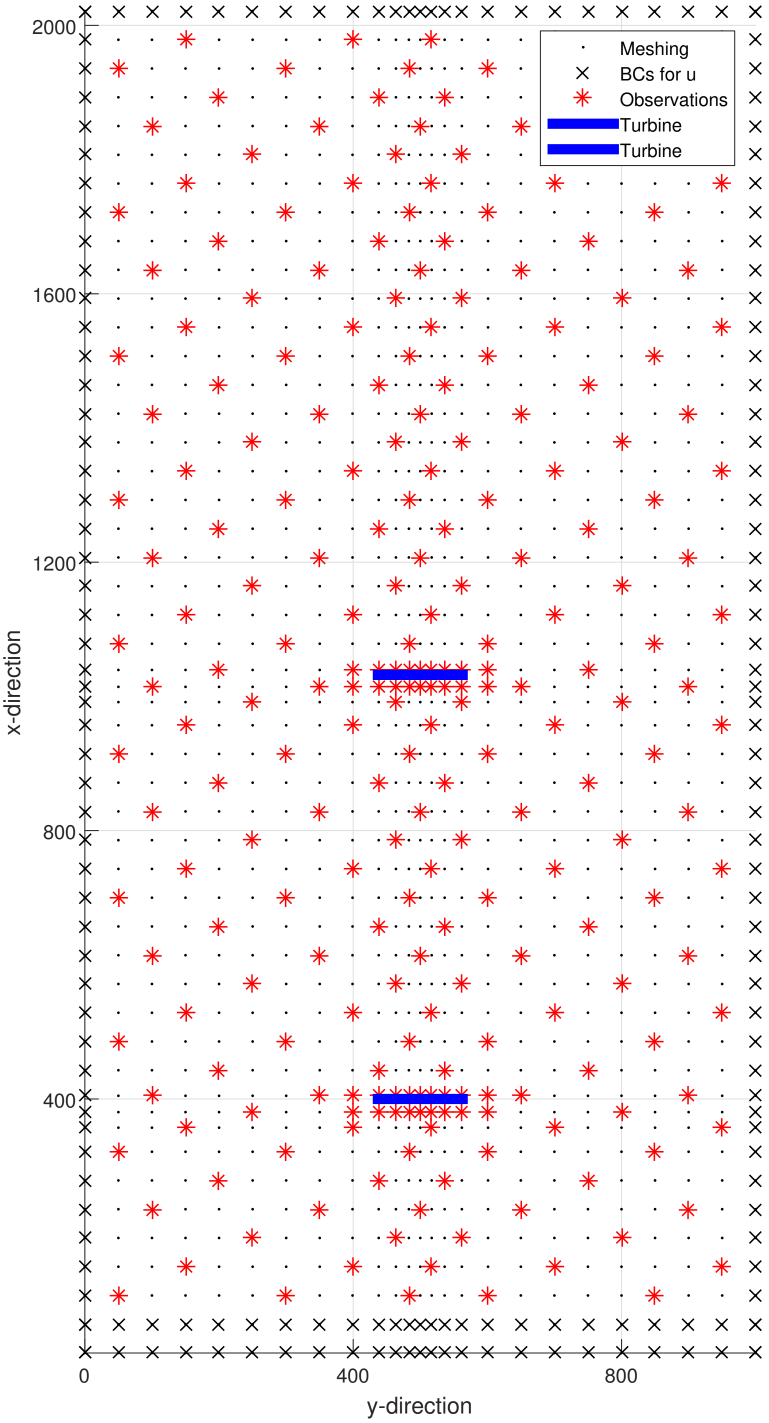
All measures are considered throughout the simulation time.

[2] Recall that the mean wake centerline is defined as the laterally averaged flow velocity throughout the simulation domain in longitudinal direction, according to Figure 2-12.



**(a)** Location of longitudinal velocity measurements for the 50x100 meshing. $11\%$ of the system outputs (long. and lat. velocities) are used as measurements fed into the observer.

**Figure 4-1:** Grid layout, turbine locations, boundary conditions and measurement locations fed into the EnKF and ApKF algorithms, used for state reconstruction. Measurements are spaced equidistantly throughout the grid, and near the turbines. The measurements are disturbed with Gaussian noise $\sigma = 0.10$ m/s.



**(b)** Location of longitudinal velocity measurements for the 25x50 meshing. $23\%$ of the system outputs (long. and lat. velocities) are used as measurements fed into the observer.

**Figure 4-1:** Grid layout, turbine locations, boundary conditions and measurement locations fed into the EnKF and ApKF algorithms, used for state reconstruction. Measurements are spaced equidistantly throughout the grid, and near the turbines. The measurements are disturbed with Gaussian noise $\sigma = 0.10$ m/s.

## 4-3 Observer tuning

For observer tuning, the internal model (WFSim) as discussed in Chapter 2 was left intact to limit the degrees of freedom. Also, this flow model already shows good performance for this specific simulation case in absence of an observer, further validating this decision. Now, the observers were tuned to more explicitly track the dynamics of SOWFA displayed in Figure 2-14. Please note that this may not be the way in which the observers will be operated in a true wind farm. However, the objective in this thesis is to push the limits of the filtering algorithms, and assess their capacity in tracking both short- and long-term fluctuations in the flow.

In quantitative terms, this means increasing the VAF, increasing the QOF and decreasing the RMS error with SOWFA data. To achieve this, the process noise covariance matrix is steadily increased and/or the measurement noise covariance matrix decreased, to be more responsive to measurement data. However, this may lead to an increase in RMS error. A trade-off is to be made.

Note that computational effort is hard to influence for a regular Kalman filter. However, for the EnKF, the number of ensemble members $n_e$ has a direct influence on the calculation time. Therefore, $n_e$ was kept to a minimum while meeting the performance goals. Paired with the decision on $n_e$ goes the decision on the initial ensemble distribution – more ensemble members allows a wider initial ensemble distribution without loss of quality. Furthermore, for the ApKF, the $z$ value, determining the number of nonzero matrix entries maintained, can be changed to increase or reduce the number of floating point operations executed per iteration.

## 4-4 Approximate Kalman filtering

Tuning the ApKF according to Section 4-3, the optimal observer settings were found and listed in Table 4-1. The covariance matrices are $P_0$, $R_k$ and $Q_k$, respectively for the initial state estimation error, measurement noise and process noise. Cross-covariance between measurement and process noise $S_k$ is neglected and kept 0 for all $k$, as previously explained in Section 3-4. $P_0$, $R_k$ and $Q_k$ are all diagonal matrices with entries equal to that given in Table 4-1, besides the last diagonal entry. This entry is zero, as the extended system vector is $\begin{bmatrix}\alpha_k^T & 1\end{bmatrix}^T$, with 1 being an artificial state known with 100% certainty, as previously outlined in Section 3-1.

**Table 4-1:** Optimal observer settings for the ApKF.

| **Mesh resolution** | $z$ | $P_0$ **(m/s)** | $R_k$ **(m/s)** | $Q_k$ **(m/s)** |
|---|---|---|---|---|
| 50x100 | 0.05 | 10.0 | 1.0 | 0.050 |
| 25x50 | 0.05 | 10.0 | 1.0 | 0.050 |

An iteration takes 8.5 seconds for the 50x100 mesh on average, and only 0.8 seconds for the 25x50 mesh.[3] Notably, this means currently only the 25x50 meshing meets the time restriction

[3] Note that iteration time decreases with a decreased number of observations, and therefore better observation localization should both improve performance and computational efficiency.

of $\leq 1$ s depicted in the observer requirements of Section 1-3.

The results are presented next, in the same order as in Section 2-3.

**Flow fields** Snapshots of the flow field at various time instants are displayed in Figure 4-2. The first phenomenon to note is the longitudinal streaky lines in the estimated flow fields for the 50x100 meshing. These erroneous flow lines may result from a poorly estimated velocity at grid points upstream, which then further propagates downstream. A number of reasons are looked at that could possibly explain this effect. Firstly, it was suggested that sparsifications made in the ApKF were perhaps no longer valid at higher resolution meshings. However, this was shown not to be the source of error, as simulations with the traditional KF yield similar problematic results. Furthermore, small modifications to WFSim initially implemented for computational efficiency were reversed, yet this phenomenon of erroneous vertical lines pertained at the 50x100 meshing. Furthermore, for an increasing number of measurements, these erroneous vertical lines become less present, suggesting that there may be a problem with the underlying model WFSim rather than with the filtering algorithms. As WFSim is still in development stage, this would not be unusual.[4]

Looking at the figures for the 25x50 meshing, more satisfactory results are obtained. The observer accounts for the overestimated wake width, and furthermore also captures some wake meandering. Secondly, the effect of the turbine hub and thereby the formation of two separate wake structures (rather than one large structure) is accounted for by the observer. The main point of critique is the wake depth: the SOWFA data shows a lower minimum flow velocity in the wakes. This may indicate model error due to WFSim incorporating a simplified rotor model, or the way in which the estimated input signal $\beta_w$ was obtained. Currently, the ApKF cannot sufficiently correct for this model error.

From this figure, the main source of error is found to be near and downstream of the second turbine, similarly to as in Section 2-3. This is in support of what can be concluded from Figure 2-9. In this figure, it was shown that there is significant variation on the estimated system input signal $\beta_w$ for turbine 2. This flow is very turbulent and the dynamics are complicated, for which WFSim is not sophisticated enough to capture. Even with observer, the estimations in this waked region are suboptimal.

Results for the lateral velocity components can be found in Appendix B.

**Maximum and mean error** The maximum and mean error over time are displayed in Figure 4-3. The average mean error over time entire simulation time is 0.41 m/s and 0.34 m/s, respectively for the 25x50 meshing and the 50x100 meshing. This is very similar to the values found for the simulation without observer (Figure 2-11). On the other hand, the average maximum errors are 6.7 m/s and 3.5 m/s for the 50x100 and the 25x50 meshing, respectively. This further suggests problematic performance for the ApKF at the 50x100 meshing, as was also noted in Figure 4-2. To further assess this statement, the mean wake centerline is analyzed.

[4] Currently, research is still ongoing to further localize the source of error. The next step is to perform simulations at different resolution meshes, as this erroneous phenomenon is not present in simulation results of the ApKF at the 25x50 meshing.

**Wake centerlines** The mean wake centerlines using the ApKF at various time instants are displayed in Figure 4-4. As the 25x50 meshing and 50x100 meshing have identical tuning parameters (Table 4-1), the results are expected to be very similar. However, note from this figure that the ApKF at 50x100 does performance worse both in terms of tracking the dynamics and the absolute fit, as expected from the previous discussions in this section. This statement is consolidated and quantified by calculating the RMS error, VAF and QOF at each time instant, as displayed in Table 4-2.

**Table 4-2:** RMS error, VAF, and QOF between the true mean wake centerline obtained from SOWFA data and estimated mean wake centerline obtained from the ApKF.

**(a)** The ApKF at 50x100 meshing.

| **Time (s)** | **RMS (m/s)** | **VAF (%)** | **QOF (%)** |
|---|---|---|---|
| 1 | $5.80 \cdot 10^{-2}$ | 16.6 | 100 |
| 200 | $6.26 \cdot 10^{-1}$ | 82.4 | 98.8 |
| 500 | $7.48 \cdot 10^{-1}$ | 77.9 | 98.3 |
| 1000 | $9.21 \cdot 10^{-1}$ | 71.8 | 97.3 |
| 1500 | $7.43 \cdot 10^{-1}$ | 81.3 | 98.3 |
| 1999 | $8.40 \cdot 10^{-1}$ | 81.0 | 97.9 |

**(b)** The ApKF at 25x50 meshing.

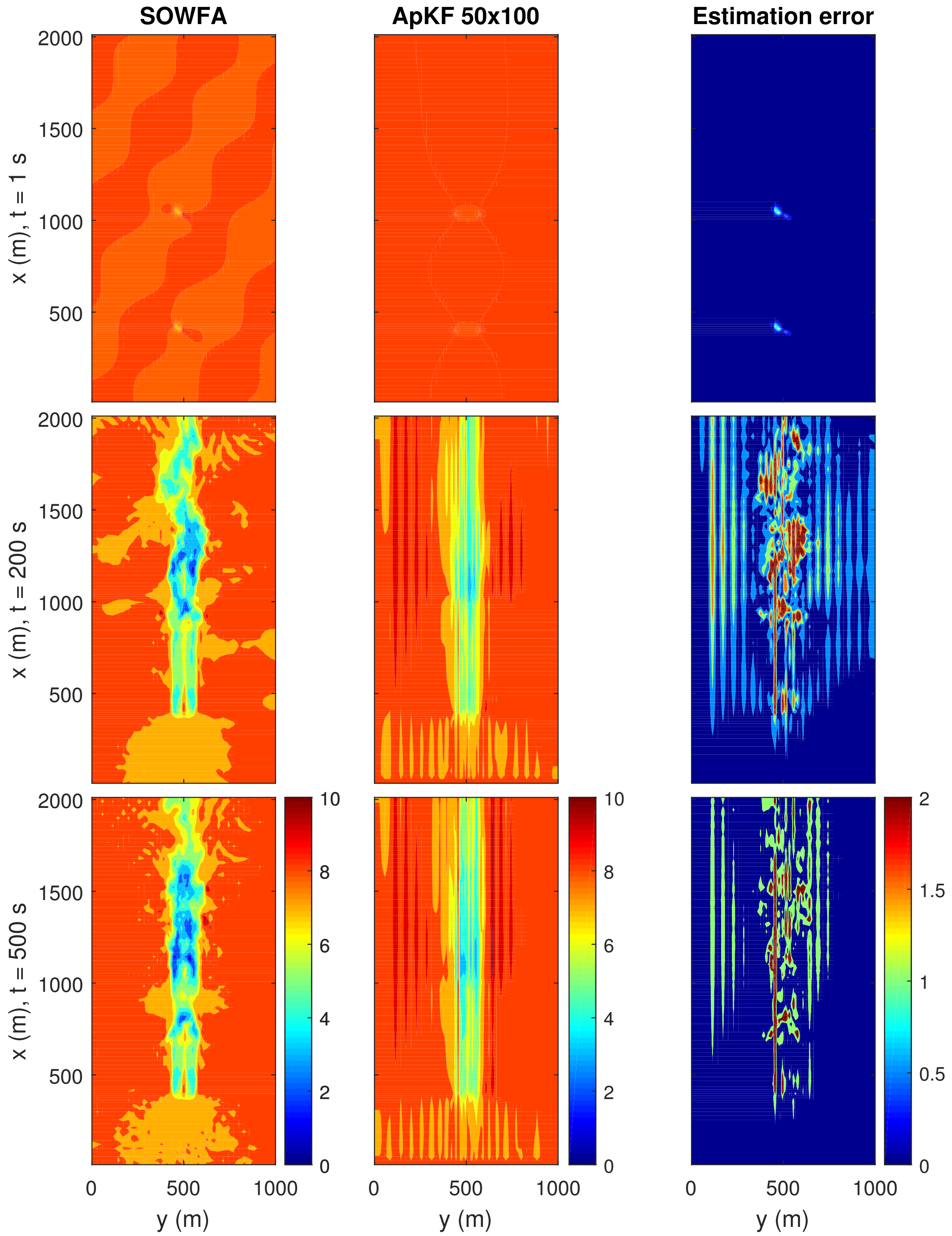
| **Time (s)** | **RMS (m/s)** | **VAF (%)** | **QOF (%)** |
|---|---|---|---|
| 1 | $4.61 \cdot 10^{-2}$ | 17.0 | 100 |
| 200 | $4.19 \cdot 10^{-1}$ | 88.8 | 99.5 |
| 500 | $5.90 \cdot 10^{-1}$ | 82.6 | 99.1 |
| 1000 | $4.33 \cdot 10^{-1}$ | 88.1 | 99.5 |
| 1500 | $4.48 \cdot 10^{-1}$ | 88.8 | 99.4 |
| 1999 | $4.14 \cdot 10^{-1}$ | 93.0 | 99.5 |

Comparing this table to Table 2-2, it can be seen that the ApKF improves measures in terms of the RMS error, the VAF and the QOF for the 25x50 meshing. On the other hand, the 50x100 meshing is worse at almost all time instants. Additional simulations have been performed at 50x100 using 21% of the measurements available instead of 11%, yet yielding no significant improvement in estimation accuracy. There are a number of explanations for this. Firstly, as mentioned before, the underlying model WFSim may be problematic at higher resolution meshes. Secondly, it is suggested that the stochasticity and model mismatch further stand out when the mesh is refined, while WFSim does not increase noticeably in fidelity. This could lead the observer astray. Alternatively, numerical errors may exist in the current implementation of both the KF and ApKF, which could explain the erroneous streaky vertical lines seen in Figure 4-2.

**Wake centerline dynamics** Finally, the mean wake centerlines over the entire simulation time can be summarized in a single figure by plotting the spatial coordinate against the temporal coordinate, displayed in Figure 4-5. This figure includes the raw SOWFA data, the ApKF at 50x100 meshing and the ApKF at 25x50 meshing. Each vertical segment corresponds to the mean wake centerline at one specific time instant. The irregularly dashed black line at $x = 400$ m indicates the control setting of the first turbine, as denoted in Figure 2-9. At the locations of the black lines $\beta_w = 0278$, and for the gaps $\beta_w = 0.153$. The regularly dotted line indicates the second turbine, operating at a constant setting $\beta_w = 0.302$. The observer attempts to correct for the underestimated wake depth in WFSim, and does so only slightly. This can be seen from the dynamics carved out slightly downstream of the second turbine, around $x = 1032$ m. Furthermore, wake recovery is corrected for by the observer, compared to the case without observer (Figure 2-10).

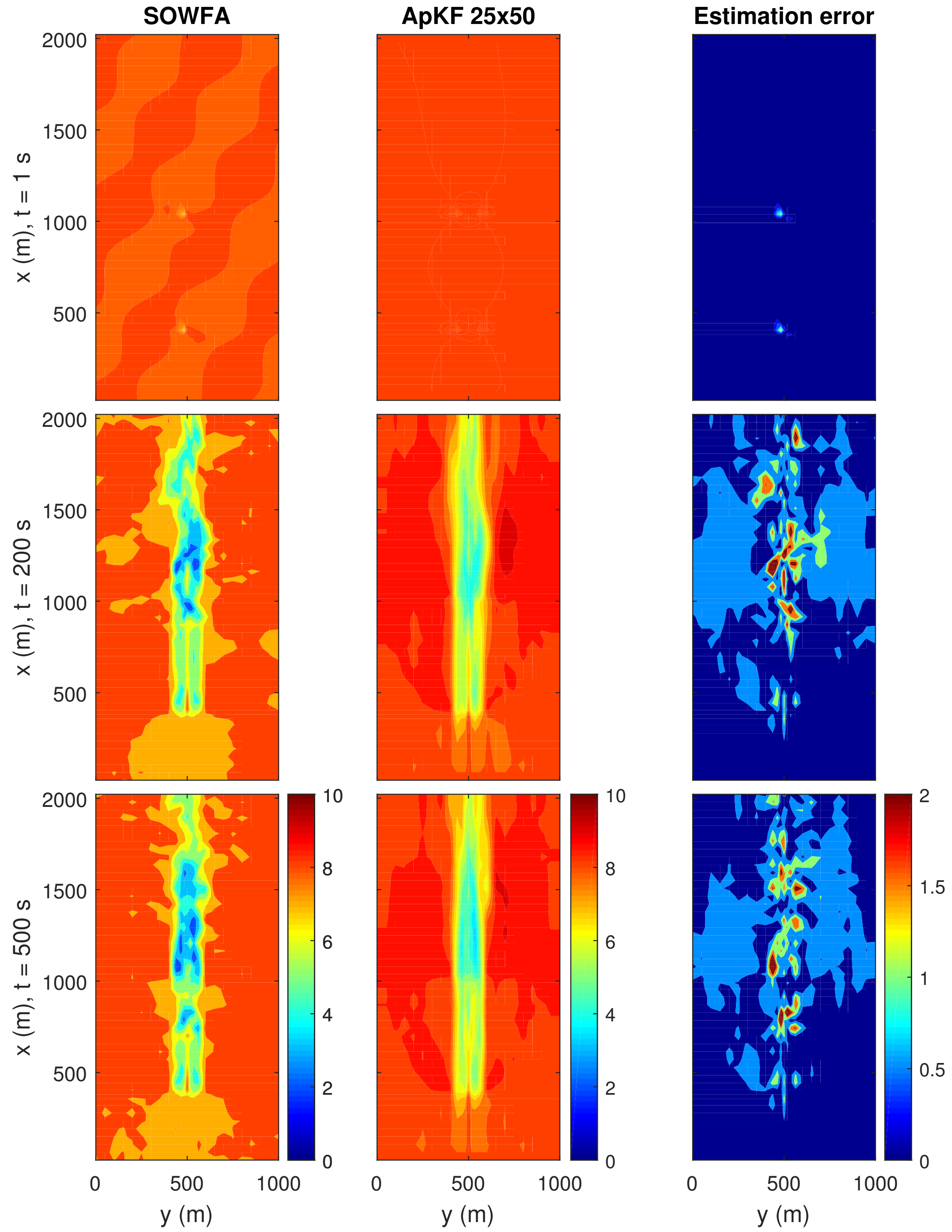
Comparing the 25x50 meshing with the 50x100 meshing, much similarity is noticed in which wake dynamics are tracked. The observer accounts for increased wake recovery and the wake dynamics are further “carved out” (corrected for) when compared to the case without observer (Figure 2-14).

**Conclusion** Reviewing Figures 4-2 to 4-5, a number of things can be noted. Firstly, the ApKF at a 25x50 meshing outperforms the ApKF at a 50x100 meshing in terms of computational cost, RMS error, VAF and QOF. The 25x50 meshing is noticeably better, accounting for an overestimated wake width, hub effects in the wake structure, wake recovery, and wake meandering effects. On the other hand, the ApKF at the 50x100 meshing is not noticeably beneficial compared to an open-loop setting; simulation in absence of an observer. A number of reasons have been suggested for this problematic performance at higher resolution meshes, of which most importantly errors in WFSim and in the numerical implementation of observers.



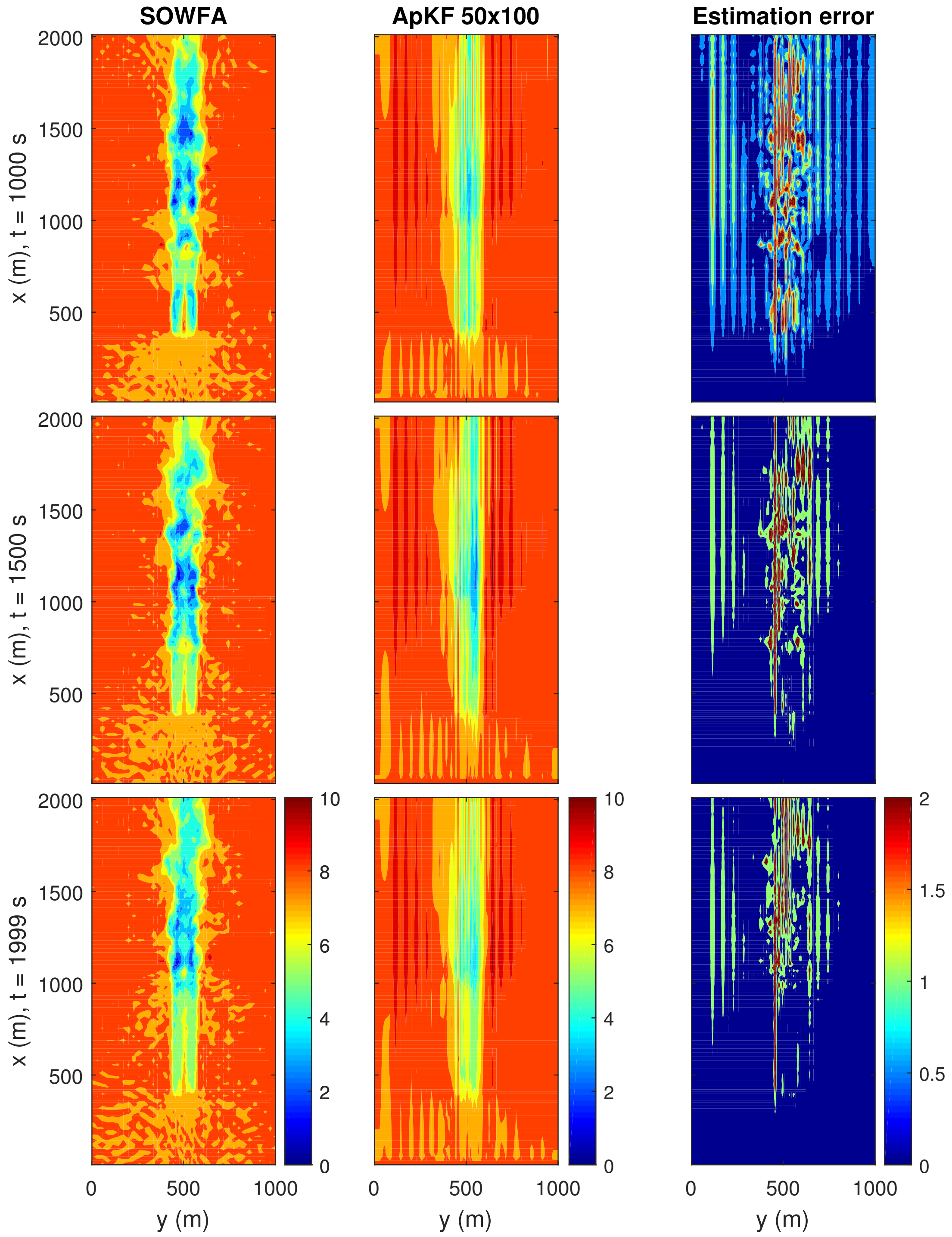
**(a)** Longitudinal flow velocity for 50x100 meshing at $t = 1, 200, 500$ s.

**Figure 4-2:** Snapshots of the longitudinal flow velocity (m/s) throughout the grid for various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s for the ApKF. The raw data is shown on the left using SOWFA data resampled at a 50x100 meshing. The ApKF is simulated under a 50x100 and 25x50 meshing.



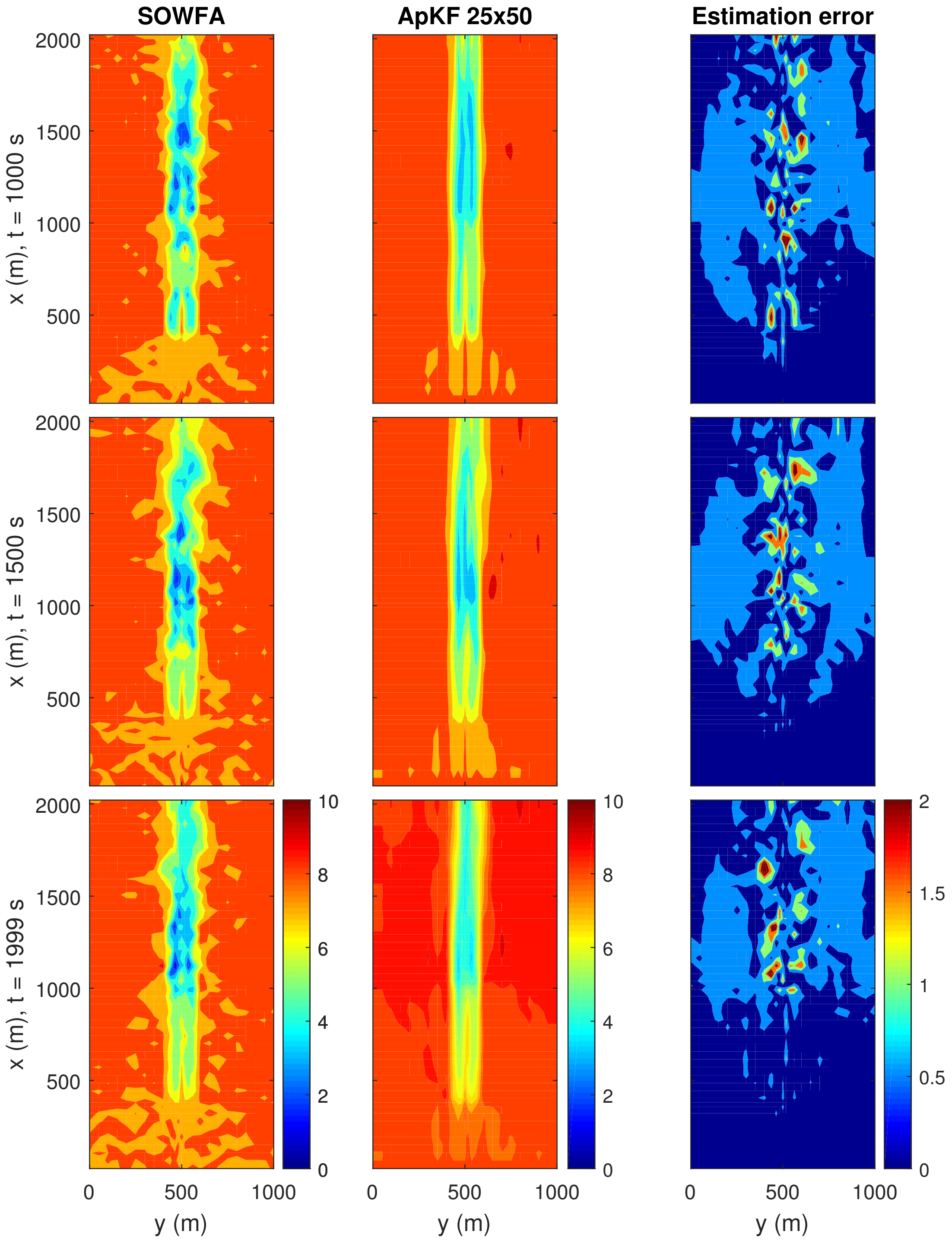
**(b)** Longitudinal flow velocity for 25x50 meshing at $t = 1, 200, 500$ s.

**Figure 4-2:** Snapshots of the longitudinal flow velocity (m/s) throughout the grid for various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s for the ApKF. The raw data is shown on the left using SOWFA data resampled at a 50x100 meshing. The ApKF is simulated under a 50x100 and 25x50 meshing.



**(c)** Longitudinal flow velocity for 50x100 meshing at $t = 1000, 1500, 1999$ s.

**Figure 4-2:** Snapshots of the longitudinal flow velocity (m/s) throughout the grid for various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s for the ApKF. The raw data is shown on the left using SOWFA data resampled at a 50x100 meshing. The ApKF is simulated under a 50x100 and 25x50 meshing.



**(d)** Longitudinal flow velocity for 25x50 meshing at $t = 1000, 1500, 1999$ s.

**Figure 4-2:** Snapshots of the longitudinal flow velocity (m/s) throughout the grid for various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s for the ApKF. The raw data is shown on the left using SOWFA data resampled at a 50x100 meshing. The ApKF is simulated under a 50x100 and 25x50 meshing.

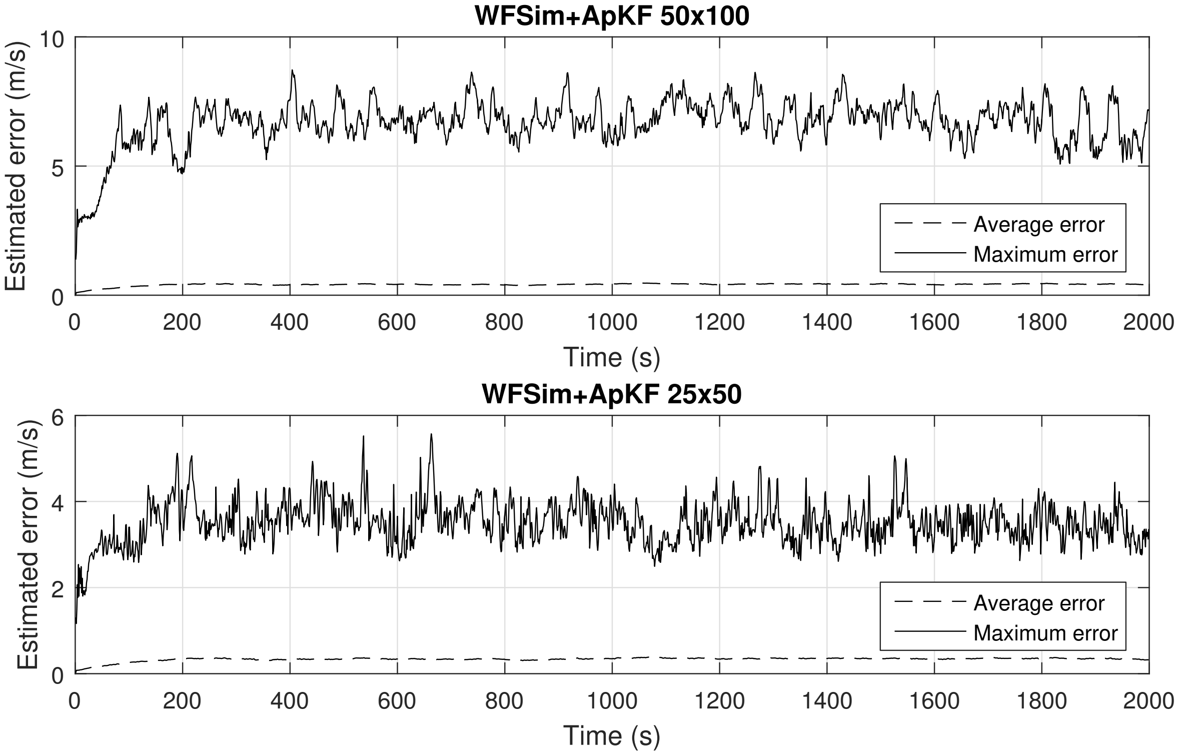


**Figure 4-3:** Error between SOWFA and the ApKF over time (m/s) for both the 50x100 meshing and the 25x50 meshing. The errors in this figure are calculated from both lateral and longitudinal velocities, spanning the entire flow field.

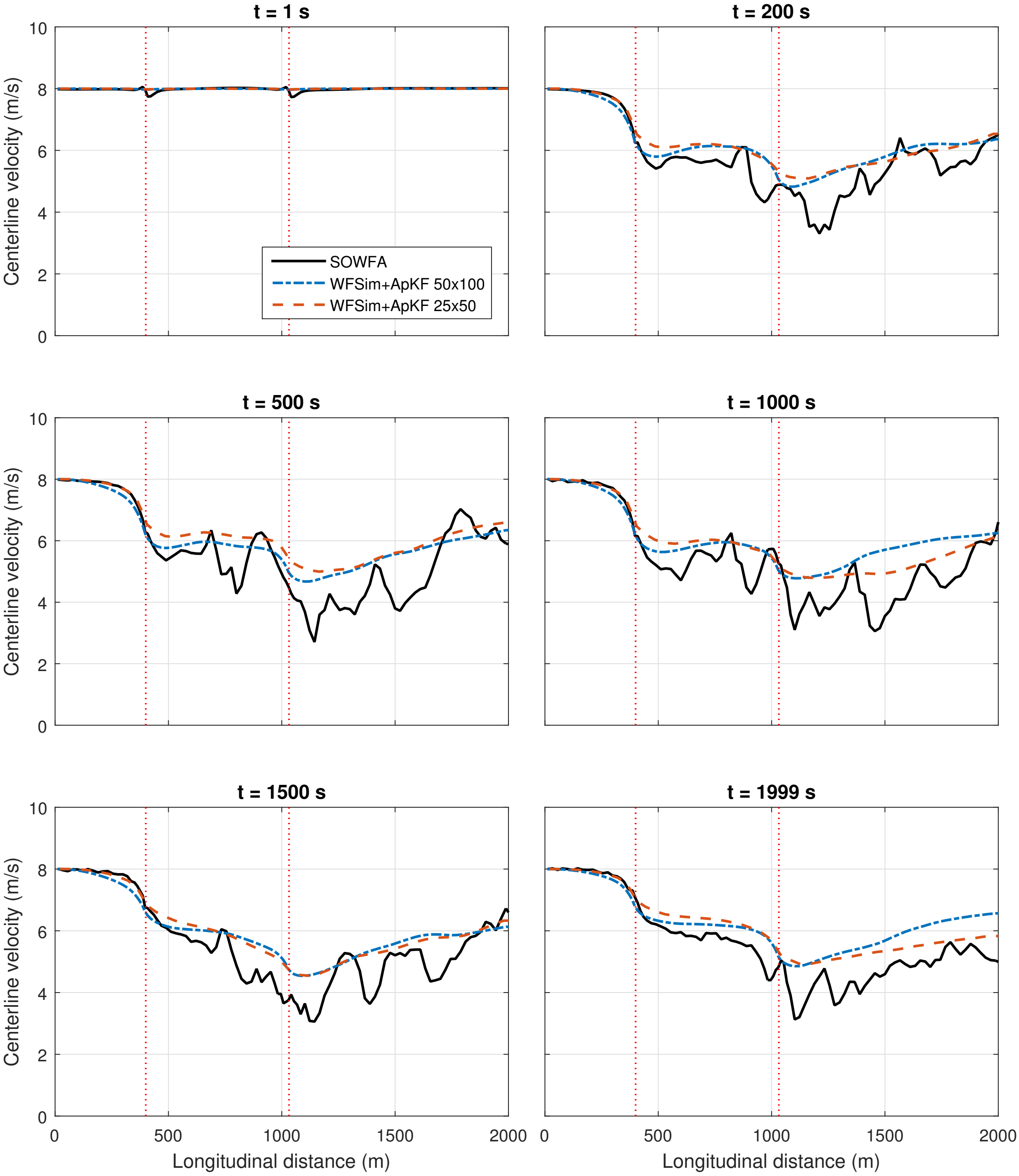


**Figure 4-4:** Mean wake centerlines (m/s) at various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s comparing SOWFA data, the ApKF at 50x100 meshing, and the ApKF at 25x50 meshing. The dotted vertical lines represent the locations of turbine 1 ($x = 400$ m) and turbine 2 ($x = 1032$ m).

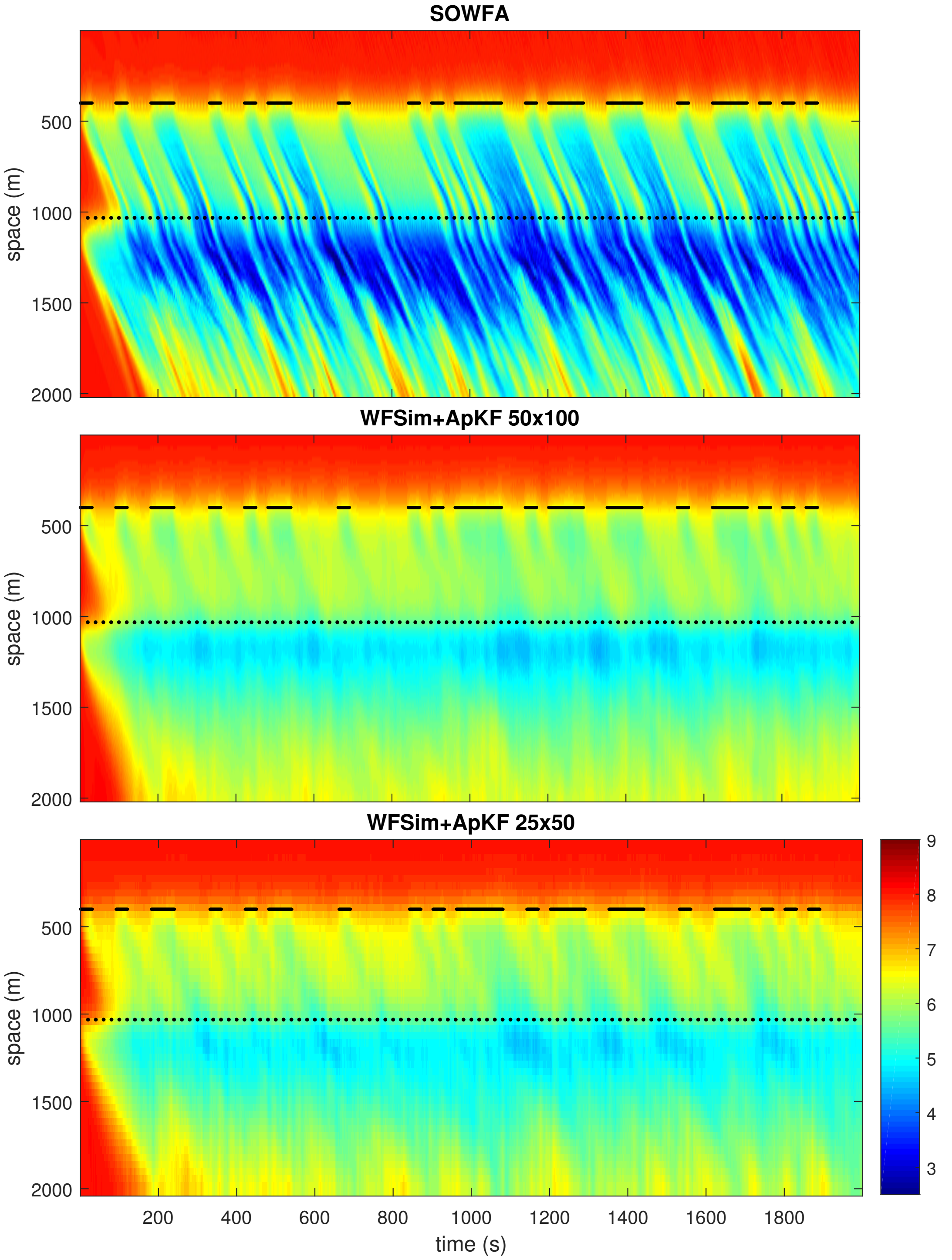


**Figure 4-5:** Mean wake centerline wind speed (m/s) for the raw SOWFA data, and the resulting ApKF simulations at the two meshes for the entire simulation time. Vertical segments represent the mean wake centerline at a certain time instant. The irregularly black dashed line shows the switching signal of the pitch angle. For the gaps $\beta_w = 0.153$ and for the black lines $\beta_w = 0.278$. Similarly, the regularly dotted line shows the location of turbine 2.

## 4-5 Ensemble Kalman filtering

Tuning the EnKF according to Section 4-3, the optimal observer settings were found to be according to Table 4-3. The covariance matrices are $W_e$, $R_e$ and $Q_e$, respectively for the initial distribution of states (representative of the true initial state covariance $P_0$), measurement noise and process noise. They are all diagonal matrices with entries equal to that given in Table 4-3.

**Table 4-3:** Optimal observer settings for the EnKF.

| **Mesh resolution** | $n_e$ | $W_e$ **(m/s)** | $R_e$ **(m/s)** | $Q_e$ **(m/s)** |
|---|---|---|---|---|
| 50x100 | 500 | 10.0 | 0.30 | 1.00 |
| 25x50 | 200 | 5.0 | 0.10 | 1.00 |

An iteration takes 8.0 seconds for the 50x100 mesh on average, and only 0.7 seconds for the 25x50 mesh, both marginally faster than the ApKF.[5] Again, only the 25x50 meshing meets the $\leq 1.0$ s time restriction as depicted in the observer performance goals in Section 1-3.

The results are presented next, in the same order as in Sections 2-3 and 4-4.

**Flow fields** Snapshots of the flow field at various time instants are displayed in Figure 4-6, for both the 50x100 meshing and the 25x50 meshing. It can clearly be noted that the EnKF algorithm relies on introducing artificial noise to the measurements and process: the flow fields are significantly less homogeneous than for the validation case (Figure 2-10) or for the ApKF (Figure 4-2). This introduces estimation errors also in the freestream flow, which initially did not even require any correction by the observer. However, the strength in this approach lies in its ability to deal with nonlinear dynamics. As such, the width in wake has been adjusted according to the SOWFA measurement data, and there is increased wake recovery at the end (upper part) of the simulation domain. Similar to as found in the ApKF, wake meandering and turbine hub effects are corrected for by the EnKF. Moreover, the EnKF does a better job at correcting for the rotor modeling problems, and is able to further accounts for the deeper wake effects found in SOWFA. To further consolidate this statement, the estimated mean wake centerlines have to be looked at.

Additionally, the 25x50 meshing appears to outperform the 50x100 meshing, as was also noted in the ApKF case. Quantitative measures will be looked at in the next paragraph to further consolidate this statement. Furthermore, as seen in all previous simulation results, the region near and behind turbine 2 remains the most troublesome in terms of providing reliable flow estimations.

Results for the lateral velocity components can be found in Appendix B.

**Maximum and mean error** The maximum and mean estimation error over time is displayed in Figure 4-7. The average error for the 50x100 grid is approximately 0.58 m/s, while for the 25x50 grid this is 0.44 m/s. Interestingly, both values are higher than in the absence of an observer (Figure 2-11) and than for the ApKF (Figure 4-3). The main reason for this is

[5] Note that iteration time decreases with a decreased number of observations, and therefore better observation localization should both improve performance and computational efficiency.

expected to be the stochasticity that is fundamental to the EnKF algorithm, as previously noted in Figure 4-6. By artificially disturbing the process and measurements in each ensemble member, more noise is introduced. However, this should allow for a better state reconstruction in nonlinear systems on average. Secondly, the fact that the higher resolution mesh has a larger average error indicates problematic performance as also seen in the ApKF.

**Wake centerlines** More insightful are the mean wake centerlines at various time instants, as displayed in Figure 4-8. This truly shows how the observer attempts to correct and track the dynamics present in the high fidelity dataset, in contrast to a more transient response in the absence of an observer (Figure 2-13). Notably, the velocity at turbine locations ($x = 400$ m and $x = 1032$ m) are very well predicted due to an increased number of measurements near these positions (as displayed in Figure 4-1). As suggested by Figure 4-6, the observer accounts for the underestimated wake depth of WFSim, which is less the case for the ApKF.

Also, wake recovery is significantly improved using the observer. The resulting RMS error, VAF values and QOF values for both simulations are listed in Table 4-4.

**Table 4-4:** RMS error, VAF, and QOF between the true mean wake centerline obtained from SOWFA data and estimated mean wake centerline obtained from the EnKF.

**(a)** The EnKF at 50x100 meshing.

| **Time (s)** | **RMS (m/s)** | **VAF (%)** | **QOF (%)** |
|---|---|---|---|
| 1 | $5.80 \cdot 10^{-2}$ | 16.6 | 100 |
| 200 | $4.73 \cdot 10^{-1}$ | 88.7 | 99.4 |
| 500 | $5.44 \cdot 10^{-1}$ | 87.9 | 99.1 |
| 1000 | $6.03 \cdot 10^{-1}$ | 88.0 | 98.9 |
| 1500 | $5.93 \cdot 10^{-1}$ | 88.4 | 98.9 |
| 1999 | $4.88 \cdot 10^{-1}$ | 94.6 | 99.3 |

**(b)** The EnKF at 25x50 meshing.

| **Time (s)** | **RMS (m/s)** | **VAF (%)** | **QOF (%)** |
|---|---|---|---|
| 1 | $4.61 \cdot 10^{-2}$ | 17.2 | 100 |
| 200 | $3.74 \cdot 10^{-1}$ | 88.1 | 99.6 |
| 500 | $4.38 \cdot 10^{-1}$ | 88.5 | 99.5 |
| 1000 | $3.67 \cdot 10^{-1}$ | 90.7 | 99.6 |
| 1500 | $4.15 \cdot 10^{-1}$ | 89.2 | 99.5 |
| 1999 | $3.15 \cdot 10^{-1}$ | 93.5 | 99.7 |

Comparing these results to Table 2-2, it can be seen that the observer at the 25x50 meshing improves flow estimates through a lower RMS error, a higher VAF and a higher QOF at all time instants. The 50x100 meshing has a better VAF at all time instants, but shows worse results in terms of RMS error and QOF. The 25x50 meshing outperforms the 50x100 meshing in computational time, RMS error, VAF and QOF. A similar relative performance was seen in the ApKF. Also, when comparing these results to Table 4-2, it is seen that the EnKF outperforms the ApKF at both resolutions in terms of RMS error and VAF and computational cost, while performing similarly in QOF.

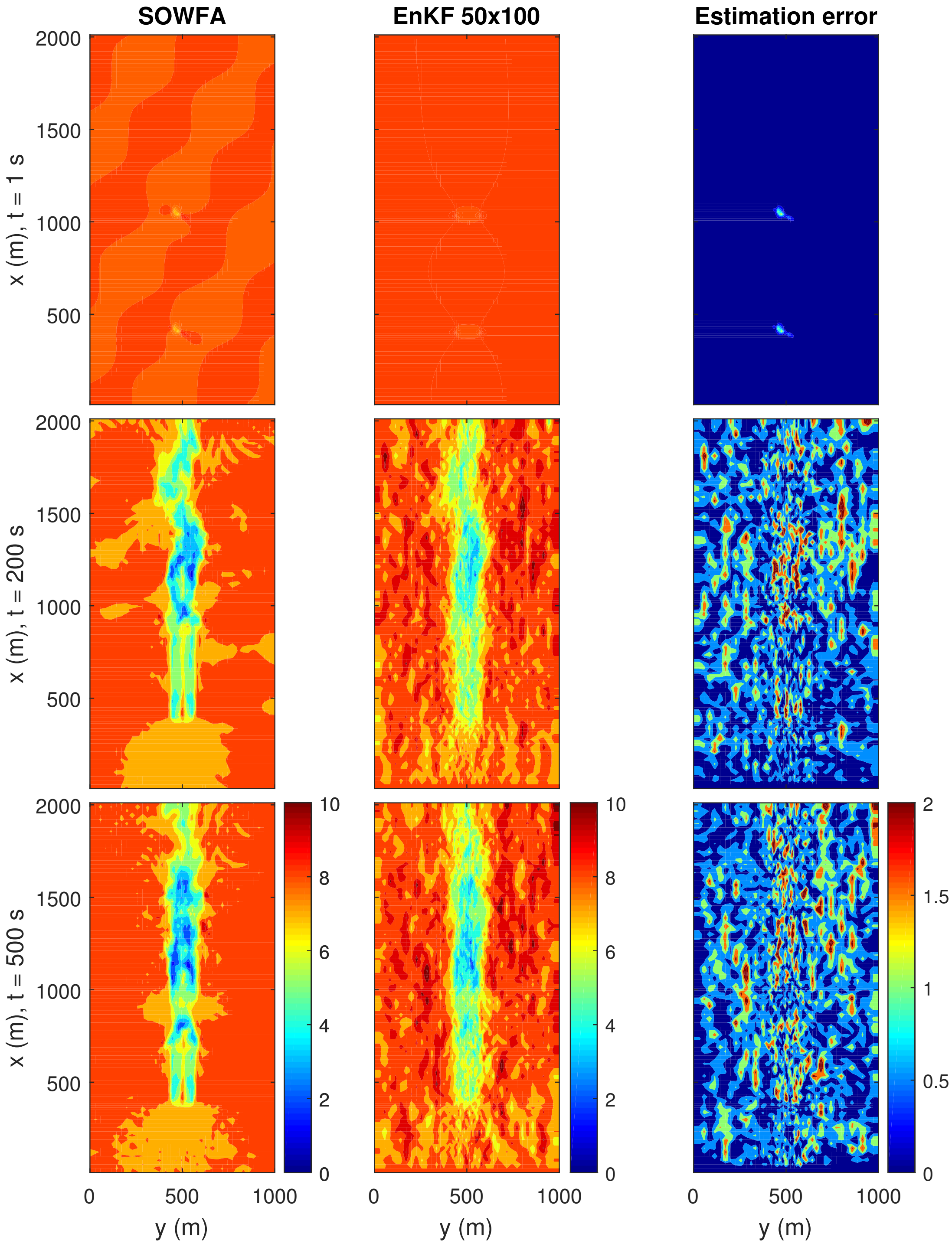
**Wake centerline dynamics** Finally, the mean wake centerlines over the entire simulation time can be summarized in a single figure by plotting the spatial coordinate against the temporal coordinate, displayed in Figure 4-9. This figure includes the raw SOWFA data, the EnKF at 50x100 meshing and the EnKF at 25x50 meshing. Each vertical segment corresponds to the mean wake centerline at one specific time instant. The irregularly dashed black line at $x = 400$ m indicates the control setting of the first turbine, as denoted in Figure 2-9. At the locations of the black lines $\beta_w = 0278$, and for the gaps $\beta_w = 0.153$. The regularly dotted line at $x = 1032$ m indicates the second turbine, operating at a constant setting $\beta_w = 0.302$. This figure more clearly shows how the EnKF is able to correct for the underestimated wake effects near and behind turbine 2. Furthermore, this figure shows the increased wake recovery accounted for. When compared to Figure 4-5 and Figure 2-14, the EnKF appears to best carve out the unmodeled dynamics in the flow.

**Conclusion** Reviewing Figures 4-6 to 4-9, it can be concluded that the EnKF at a 25x50 meshing does very well at correcting for unmodeled dynamics in reconstructing flow fields. Phenomena corrected for include an overestimated wake width, underestimated wake depth, turbine hub effects, wake meandering and wake recovery. Furthermore, this observer performs iterations in about 0.7 seconds, making it applicable for real-time control. Quantitatively, this filter reduces the RMS error and increases the VAF significantly for the estimated mean wake centerline.

The EnKF at a 50x100 meshing performs worse than its 25x50 meshing counterpart. Similar results were noted in the ApKF, further suggesting that the problem may lie in the underlying flow model WFSim, and not in the filtering algorithms. With WFSim still in development, this would not be unusual. Furthermore, it may be that the increase in measured dynamics due to an increased spatial resolution lead the observer astray. More fundamental numerical issues in the observer implementation could also cause these erroneous results, but it is unlikely since both the ApKF and EnKF show problematic results while relying on fundamentally different algorithms. Due to time restrictions, the current results suffice for demonstration purposes, and as a proof of concept.

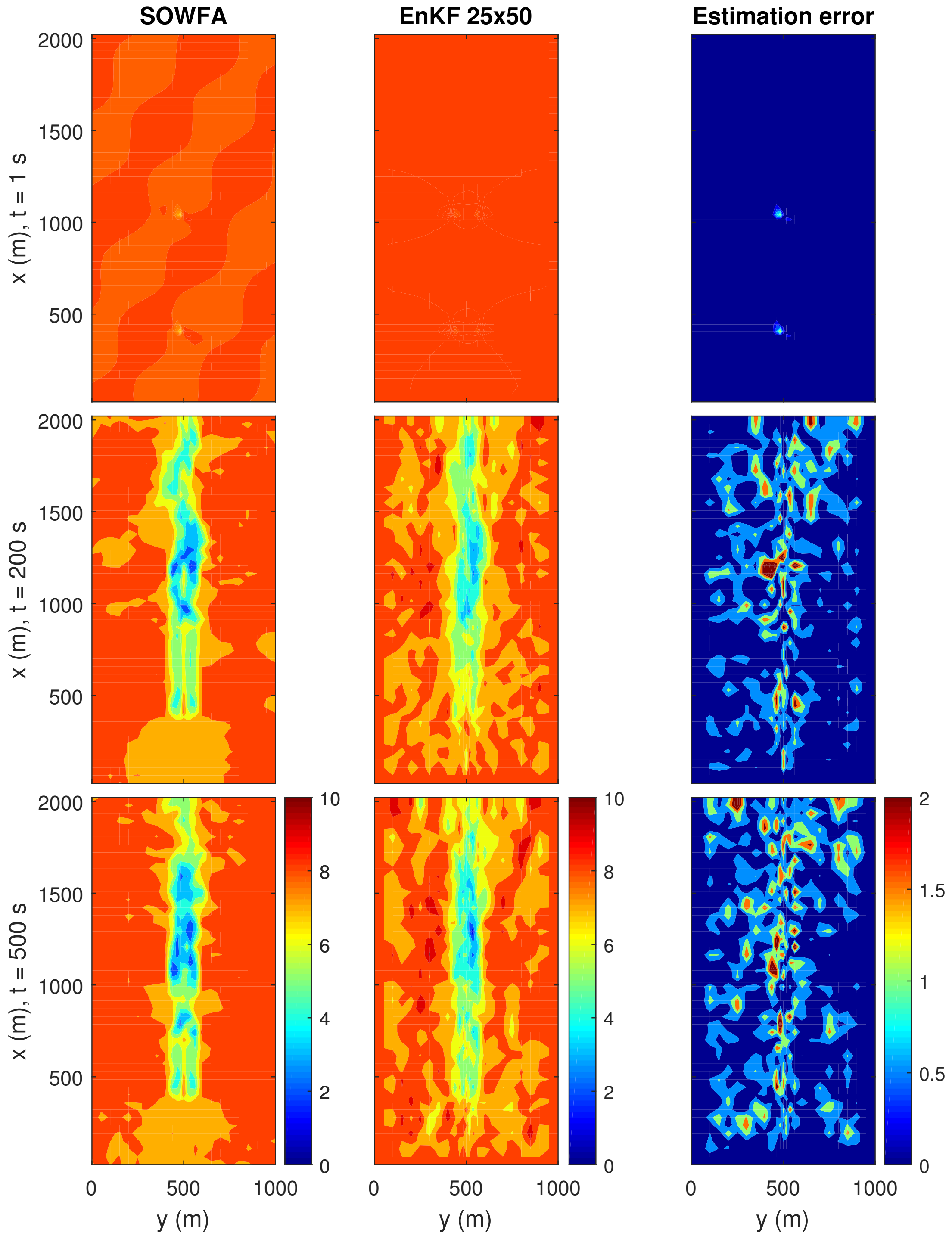
Furthermore, the mean estimation error for the ApKF is smaller than that of the EnKF according to Figure 4-3. However, it typically has a higher RMS error and lower VAF in terms of wake centerline tracking, according to Table 4-2. This may indicate that the ApKF focuses more on tracking the overall dynamics, while the EnKF performs better in terms of the area of interest – the turbine wakes.

For current use, the EnKF at 25x50 meshing is strongly recommended over the 50x100 meshing, for both computational benefits and for better state reconstruction performance. As the domain size becomes larger, more sophisticated turbulence models are implemented, or less turbulent measurement data is used, the 50x100 meshing may become more interesting for implementation purposes. That is, assuming the source of error in WFSim can be located and resolved.



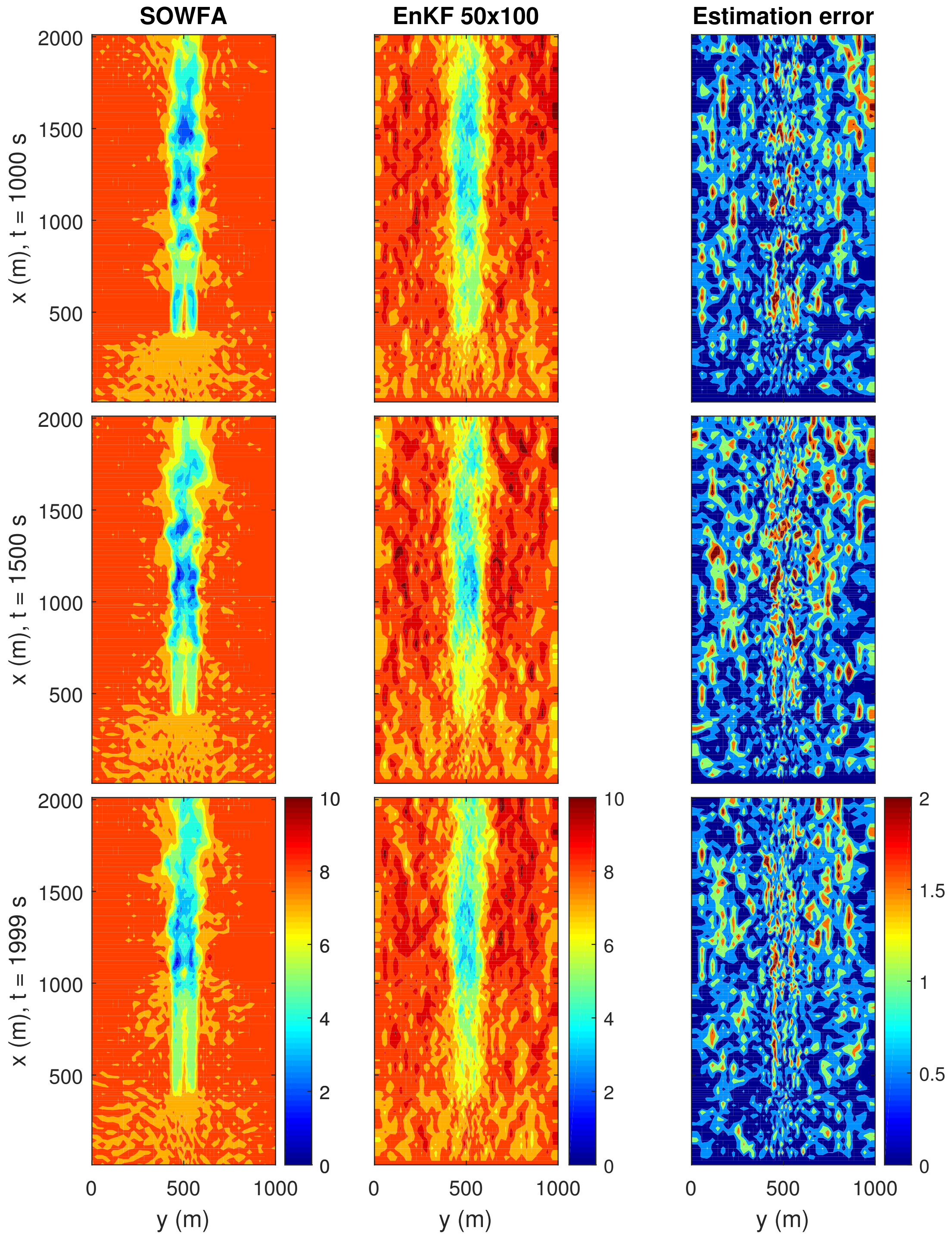
**(a)** Longitudinal flow velocity for 50x100 meshing at $t = 1, 200, 500$ s.

**Figure 4-6:** Snapshots of the longitudinal flow velocity (m/s) throughout the grid for various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s for the EnKF. The raw data is shown on the left using SOWFA data resampled at a 50x100 meshing. The EnKF is simulated under a 50x100 and 25x50 meshing.



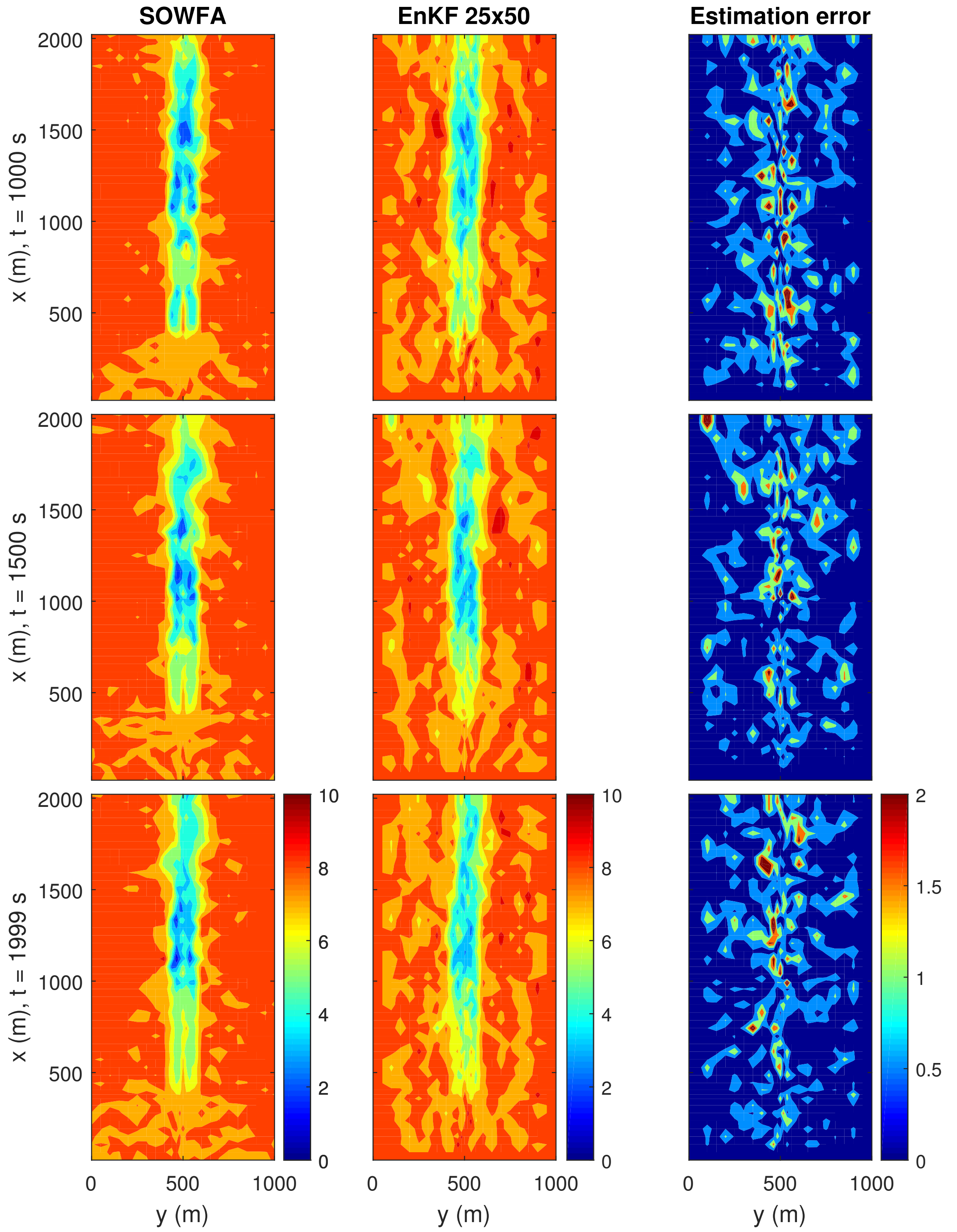
**(b)** Longitudinal flow velocity for 25x50 meshing at $t = 1, 200, 500$ s.

**Figure 4-6:** Snapshots of the longitudinal flow velocity (m/s) throughout the grid for various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s for the EnKF. The raw data is shown on the left using SOWFA data resampled at a 50x100 meshing. The EnKF is simulated under a 50x100 and 25x50 meshing.



**(c)** Longitudinal flow velocity for 25x50 meshing at $t = 1000, 1500, 1999$ s.

**Figure 4-6:** Snapshots of the longitudinal flow velocity (m/s) throughout the grid for various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s for the EnKF. The raw data is shown on the left using SOWFA data resampled at a 50x100 meshing. The EnKF is simulated under a 50x100 and 25x50 meshing.



**(d)** Longitudinal flow velocity for 50x100 meshing at $t = 1000, 1500, 1999$ s.

**Figure 4-6:** Snapshots of the longitudinal flow velocity (m/s) throughout the grid for various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s for the EnKF. The raw data is shown on the left using SOWFA data resampled at a 50x100 meshing. The EnKF is simulated under a 50x100 and 25x50 meshing.

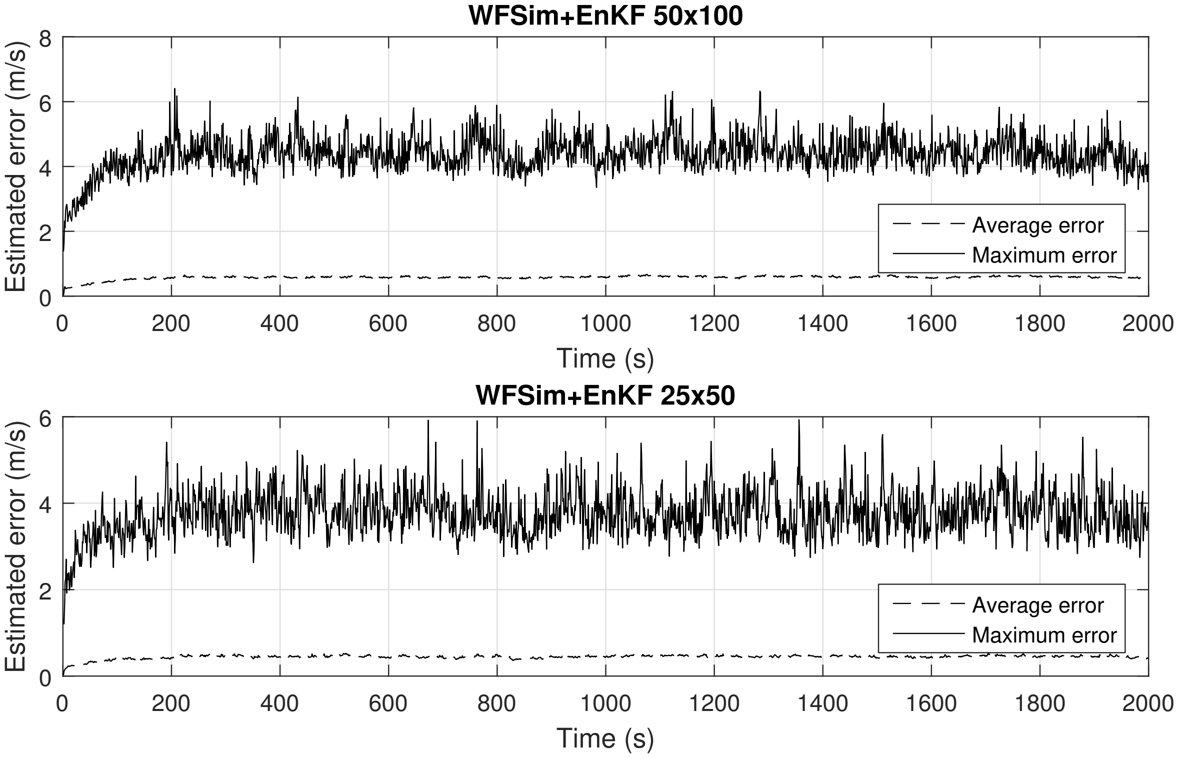


**Figure 4-7:** Error between SOWFA and the EnKF over time (m/s) for both the 50x100 meshing and the 25x50 meshing. The errors in this figure are calculated from both lateral and longitudinal velocities, spanning the entire flow field.

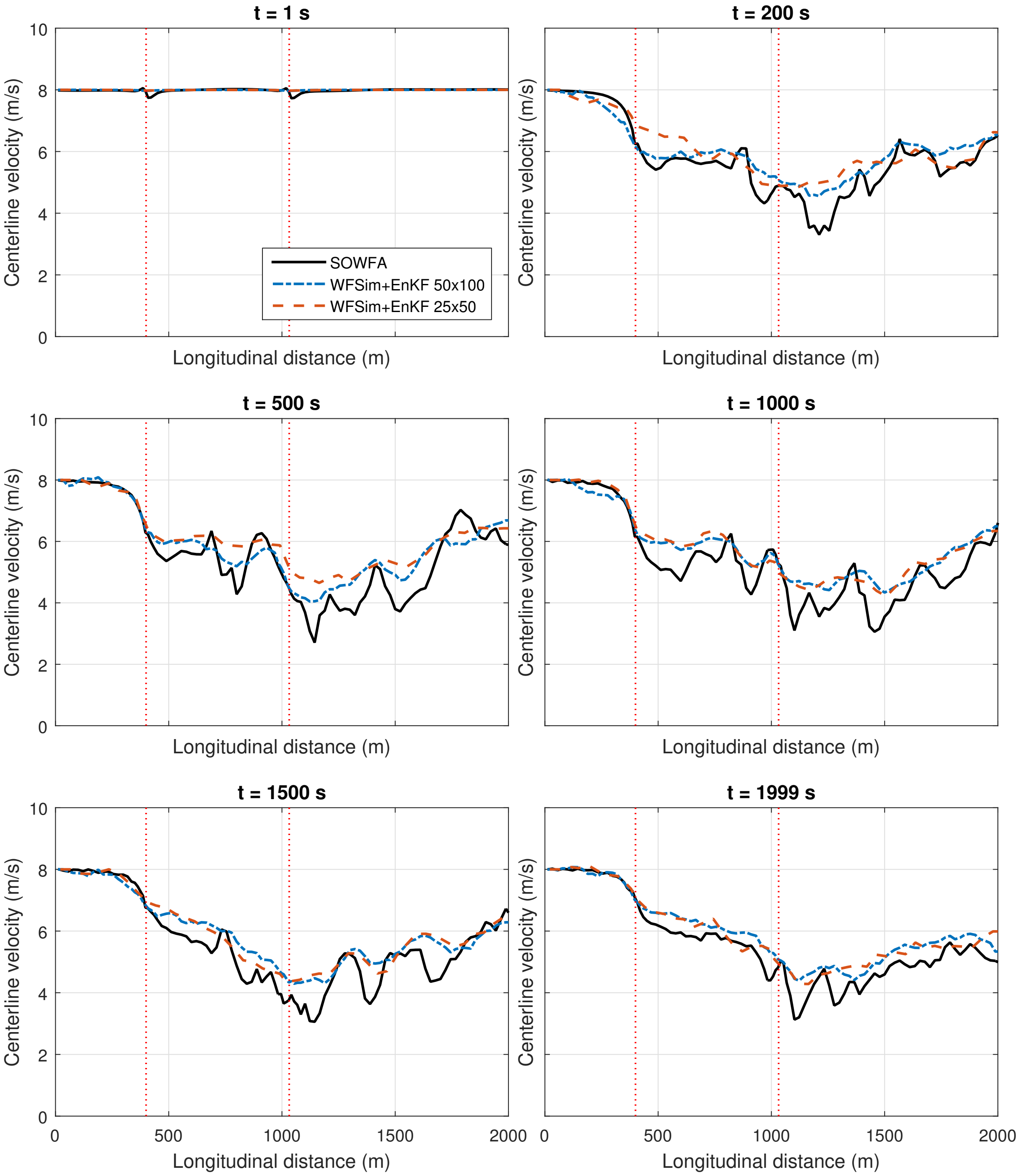


**Figure 4-8:** Mean wake centerlines (m/s) at various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s comparing SOWFA data, the EnKF at 50x100 meshing, and the EnKF at 25x50 meshing. The dotted vertical lines represent the locations of turbine 1 ($x = 400$ m) and turbine 2 ($x = 1032$ m).

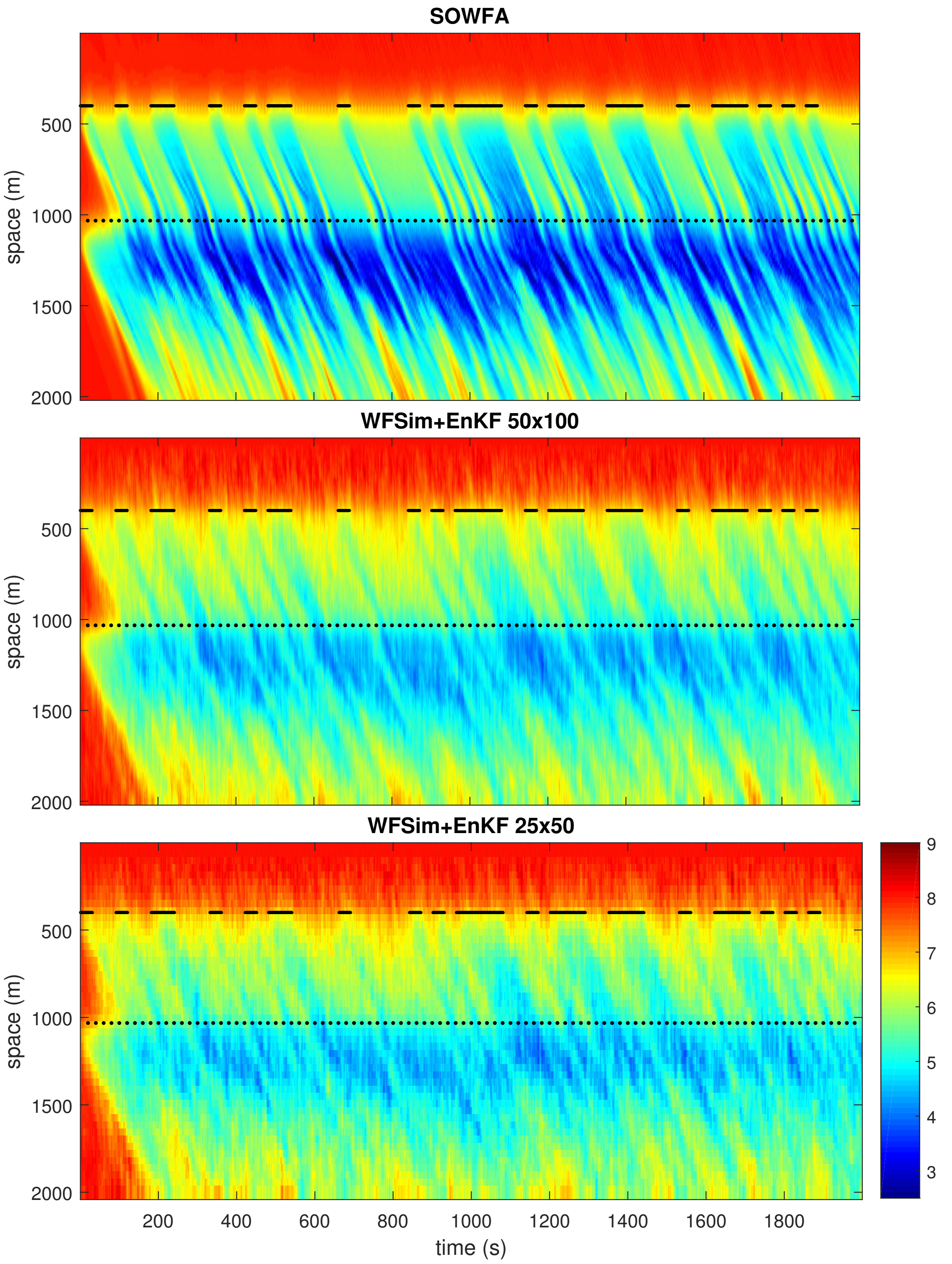


**Figure 4-9:** Mean wake centerline wind speed (m/s) for the raw SOWFA data, and the resulting EnKF simulations at the two meshes for the entire simulation time. Vertical segments represent the mean wake centerline at a certain time instant. The irregularly black dashed line shows the switching signal of the pitch angle. For the gaps $\beta_w = 0.153$ and for the black lines $\beta_w = 0.278$. Similarly, the regularly dotted line shows the location of turbine 2.

## 4-6 Overview and comparison

The RMS error, VAF and QOF for the mean wake centerlines in the open-loop, validation case (WFSim), the ApKF, and the EnKF are repeated in Table 4-5 for both the 50x100 meshing and the 25x50 meshing, respectively. This table gives a clear overview and allows for easy comparison between the observers.[6]

**Table 4-5:** Overview of the RMS errors, VAFs and QOFs between the true mean wake centerline obtained from SOWFA data, for the simulation without observer, and for the simulations with the EnKF and with the ApKF.

**(a)** High resolution meshing: 50x100.

| | **RMS (m/s)** | | | **VAF (%)** | | | **QOF (%)** | | |
|---|---|---|---|---|---|---|---|---|---|
| **Time (s)** | **WFSim** | **EnKF** | **ApKF** | **WFSim** | **EnKF** | **ApKF** | **WFSim** | **EnKF** | **ApKF** |
| 1 | 0.058 | 0.058 | 0.058 | 16.8 | 16.6 | 16.6 | 100 | 100 | 100 |
| 200 | 0.460 | 0.473 | 0.626 | 85.0 | 88.7 | 82.4 | 99.4 | 98.8 | 99.4 |
| 500 | 0.711 | 0.544 | 0.748 | 73.5 | 87.9 | 77.9 | 98.5 | 98.3 | 99.1 |
| 1000 | 0.576 | 0.603 | 0.921 | 82.1 | 88.0 | 71.8 | 99.0 | 97.3 | 98.9 |
| 1500 | 0.669 | 0.593 | 0.743 | 77.5 | 88.4 | 81.3 | 98.6 | 98.3 | 98.9 |
| 1999 | 0.372 | 0.488 | 0.840 | 92.0 | 94.6 | 81.0 | 99.6 | 97.9 | 99.3 |

**(b)** Lower resolution meshing: 25x50.

| | **RMS (m/s)** | | | **VAF (%)** | | | **QOF (%)** | | |
|---|---|---|---|---|---|---|---|---|---|
| **Time (s)** | **WFSim** | **EnKF** | **ApKF** | **WFSim** | **EnKF** | **ApKF** | **WFSim** | **EnKF** | **ApKF** |
| 1 | 0.046 | 0.046 | 0.046 | 17.2 | 17.2 | 17.0 | 100 | 100 | 100 |
| 200 | 0.491 | 0.374 | 0.419 | 85.3 | 88.1 | 88.8 | 99.4 | 99.5 | 99.6 |
| 500 | 0.836 | 0.438 | 0.590 | 64.6 | 88.5 | 82.6 | 98.1 | 99.1 | 99.5 |
| 1000 | 0.642 | 0.367 | 0.433 | 77.9 | 90.7 | 88.1 | 98.8 | 99.5 | 99.6 |
| 1500 | 0.785 | 0.415 | 0.448 | 68.8 | 89.2 | 88.8 | 98.3 | 99.4 | 99.5 |
| 1999 | 0.499 | 0.315 | 0.414 | 88.6 | 93.5 | 93.0 | 99.3 | 99.5 | 99.7 |

In both terms of RMS error and VAF, the EnKF outperforms the ApKF at both resolutions. This is as expected, with EnKF typically being more suitable for nonlinear systems. However, as the EnKF algorithm relies on stochasticity, the mean and maximum estimation error of the total flow field is higher than for the ApKF. Both the EnKF and the ApKF visibly account for wake recovery, wake meandering, an overestimated wake width, and turbine hub effects. Furthermore, the EnKF noticeably accounts for an underestimated wake depth, which the ApKF does not. Only the 25x50 meshings meets the time restriction of $\leq 1.0$ s and thus allows real-time control, as depicted in the observer design goals in Section 1-3.

Interesting to note is that, irrespective of the decrease in computation time, the 25x50 observers are recommended over the 50x100 observers due to significantly better performance in state reconstruction. Moreover, the 50x100 observers are currently problematic in terms of performance, and are not ready for any kind of implementation in their current form. Because this was noted for both the ApKF and the EnKF, the source of error is suggested to be the underlying WFSim flow model. Secondly, more fundamental numerical issues in the observer implementations could also cause similar results, but is unlikely as similar problems are shown in two fundamentally different observer algorithms.

[6] In the following analysis, the QOF measure will be neglected, as it shows near ideal performance in every simulation, making it hard to compare performance with.

What is very important to conclude from this chapter is the importance of the meshing resolution. In the current implementation, tuning the observer should be secondary to choosing the right meshing size. Perhaps when the fidelity of WFSim increases (e.g., through adding the vertical dimension or/and sophisticated turbulence submodels) and errors are resolved, the higher resolution meshing may improve results and become of use. For now, lower resolution meshes are strongly recommended for implementation.

Finally, the computational cost per iteration for the absence of a filter and the Kalman filters are displayed in Table 4-6. The enhanced filters are very similar is computational cost, taking about 7 to 8 times as much computational resources compared to the simulation without observer. The regular KF takes far too much computational resources to be of any use, and scales up more-than-linearly when increasing the number of states. On the other hand, the time-efficient KF variants scale up approximately linearly with the number of states. In total, the preference goes to the EnKF, being about $5\% - 10\%$ faster than the ApKF while providing better performance in terms of RMS error and VAF for the mean wake centerline.

**Table 4-6:** Average computational cost per iteration in seconds.

| | **WFSim** | **KF** | **EnKF** | **ApKF** |
|---|---|---|---|---|
| 50x100 | 1.2 | 1125 | 8.0 | 8.5 |
| 25x50 | 0.1 | 12.8 | 0.7 | 0.8 |

With WFSim sampled at 1 Hz, both the EnKF and the ApKF at a 25x50 meshing allow real-time implementation, with a computational time less than the actual simulation time. While these observers are expected to be significantly faster in a machine code (e.g., `C` or `Fortran`) implementation and excluding the developer/debugging options, the current code shows a good proof of concept.

Due to time restrictions, the current results suffice for demonstration purposes, and as a proof of concept. The 25x50 filters perform very well and allow direct implementation, with a preference for the EnKF. The EnKF is known to better deal with nonlinearity, as confirmed by the presented results. Furthermore, it is highly parallelizable, and therefore should be more efficient as the system is scaled up.

Future improvements of WFSim, implementations of the observers in machine code, and developments in the field of ensemble Kalman filtering (or more generally, particle filtering) should further improve the results presented in this chapter.

# Chapter 5

# Conclusions, recommendations and future outlook

In this work, two time-efficient variants of the Kalman filter (KF) were depicted and assessed using a high fidelity dataset. Internally, these filtering algorithms rely on the medium fidelity WindFarmSimulator (WFSim) model developed at the Delft University of Technology (TU Delft). The high fidelity dataset used for model validation and observer assessment was obtained from Simulator fOr Wind Farm Applications (SOWFA), a highly sophisticated simulation model from the National Renewable Energy Laboratory (NREL) and TU Delft. Simulations were of a two turbine case in a 1000 by 2000 m domain, meshed at two different resolutions: 25x50 and 50x100 grid points, respectively.

First, WFSim was validated with SOWFA data, thus simulations were run in absence of a filter. Good results were shown, with WFSim predicting a mean, smoothened flow. However, WFSim does not capture all model dynamics due to its simplified turbulence model, due to the absence of the vertical dimension (2D instead of 3D), and its simplified rotor model. Namely, hub effects are neglected and wake recovery is underestimated. Also, the wake width is overestimated, possibly due to the absence of lateral turbulence in WFSim. Furthermore, wake meandering, wake skewing and wake deflection are not modeled. The most troublesome region of flow estimation is near and behind the second, downstream turbine. An observer would potentially correct for these unmodeled dynamics.

Secondly, the time-efficient KF variants were assessed, namely an Approximate Kalman filter (ApKF) and an Ensemble Kalman filter (EnKF). The ApKF relies on the traditional KF algorithm, but improves time-efficiency by enforcing a sparsification of the main model matrices and state error covariance matrix. In the EnKF, the system matrices remain intact, yet the update algorithm is simplified by replacing the covariance matrices with a sample covariance. The EnKF is often found to better deal with nonlinear dynamics (not requiring a linearized model) and high order systems with many states ($10^3 - 10^6$) and measurements. In simulations with 23% and 11% of measurements fed into the observers, respectively for the 25x50 and 50x100 meshing, good results were shown.

The observers relying on the WFSim model at a 25x50 meshing showed best performance. Many unmodeled dynamics are accounted for: the overestimated wake width, underestimated wake depth, wake recovery, wake meandering and turbine hub effects. Quantitatively, for the mean wake centerline[1] the RMS error decreased by up to 50% and VAF increased by up to 24 percent points, at certain time instants in simulation. Furthermore, the algorithms take $0.7 - 0.8$ s to iterate for WFSim with a timestep of 1 s, thereby allowing real-time closed-loop control.

Interesting to note is that the observers at the 25x50 meshing outperforms those at the 50x100 meshing, even when relying on less measurements. The main reason for this is expected to be problems in the underlying WFSim model at higher resolution meshes. Notably, from Chapter 4 it can be concluded that choosing the right mesh resolution is of invaluable importance for good observer results, in the current implementation.

In conclusion, these observers significantly improve flow fields estimations by relying internally on the WFSim model. A 25x50 meshing appears optimal over the 50x100 meshing, in terms of both accuracy and computational cost. Furthermore, the EnKF appears preferable over the ApKF due to its ability to deal with model nonlinearity and parallelizability, showing slightly better filtering results at a lower computational cost. These filtering algorithms can be used in real-time closed-loop control, as their iteration time is less than the model timestep. The work presented in this thesis is a first big step to closed-loop control of wind farms.

## Recommendations

If this work is continued, a number of recommendations are made (in order of importance).

Firstly, if this work is continued, it is strongly suggested to further look at the discrepancy in results between the 50x100 meshing and the 25x50 meshing. It seems counter-intuitive that the higher resolution meshing performs worse in all ways when compared to the lower resolution meshing, both for the ApKF as for the EnKF. The numerical implementation of WFSim, the implementation of the observers, and any simplifications (e.g., in the SVD) made in the algorithms should be reconsidered. Furthermore, different settings in WFSim are to be explored, in close collaboration with Boersma (who is currently developing the code).

Secondly, the number of observations currently fed into the observer algorithms is not realistic (Figure 4-1). The work presented in this thesis attempted to push the limits of the filtering algorithms. The number and location of measurements are to be optimized and limited to realistic values, after which more simulations are to be performed to assess performance.

Thirdly, the way in which the turbine operating settings are mapped from SOWFA to WFSim is not realistic (Section 2-3-2). Namely, an averaged streamwise flow velocity at each turbine rotor $U_D$ is needed at each time instant, which is typically not available in real wind farms. Suggested is to avoid this mapping entirely, and implement such a mapping in the WFSim model itself. This would allow one to assign physical turbine operating settings in WFSim, rather than a theoretical axial induction factor.

Fourthly, this work is limited to axial-induction-based control: the flow is excited using rotor blade pitch, while maintaining a constant turbine yaw angle. However, yaw control

[1] Defined as the laterally averaged longitudinal flow velocity throughout the domain, by Figure 2-12.

appears promising in high fidelity simulations. Therefore, it is important to assess observer performance for yaw excitation, and furthermore the implications for observer design (e.g., “is the model sparsification in the ApKF valid for different yaw angles?”). However, this does require more SOWFA simulations to be performed.

Fifthly, for the filtering algorithms specific, each a recommendation can be made. For the EnKF, the initial distribution should further be considered. Currently, the initial ensemble of particles are distributed equidistantly throughout the state space. However, this may not be the optimal initial distribution. Secondly, for the ApKF, it may be of interest to consider how the model changes under the sparsification, e.g., by looking at changes in its poles and zeros. It is also of unvaluable importance to consider the validity of the model sparsification under different inflow conditions, as true wind farms usually do not operate under a constant inflow direction. Furthermore, for both filters, the covariance matrices should be tuned in a more sophisticated manner. Instead of applying a diagonal matrix, it would make sense to put higher covariance at grid points in the flow wakes, but low covariance at grid points in the freestream. For example, this would stop the EnKF from introducing unnecessary noise to the freestream flow in flow field estimates, thereby improving results.

Sixthly, with recent work on WFSim providing linearized models of wind farms, it may be of interest to assess the performance using a fully linearized flow model. This will significantly reduce computational cost, as the system matrices and model manipulation only have to be calculated once in simulation. Note that it is important to assess these linearized observers under different inflow conditions (direction and speed).

Seventhly, for control algorithms such as predictive control, an $n$-step-ahead prediction is necessary of the flow field. A small extension should be made to the filtering algorithms to provide this $n$-step-ahead prediction of the system states, and their accuracy should be assessed through simulation.

Finally, in case simulations would like to be run in real-time, the algorithms are to be rewritten in machine code (e.g., `C`). It is expected that this will also significantly reduce the computational cost paired with the algorithms, or alternatively allow better results in equal iteration time.

## Future work

Currently, there is a chance Doekemeijer will continue (parts of) this work as a researcher at TU Delft, but this has not yet been set in stone. Furthermore, work is currently still ongoing related to the development of WFSim, which will reflect back into the filtering algorithms. Namely, Boersma is working on improving the rotor submodel to include hub effects. Furthermore, problems seem to persist in the linearized version of WFSim, which may obstruct model observability analyses and the implementation of linearized filters.

Boersma may also continue (parts of) this work, as these observers are essential in the pursuit of real-time closed-loop control of wind farms.

# Appendix A

# Supplementary background theory: wind and wind turbine control

This appendix contains a basic introduction to wind and single turbine control. If the reader is unfamiliar with the topic at hand, this is a suggested appendix to read prior to Chapters 2 to 4.

## A-1 Flow around a turbine

Wind turbines convert kinetic energy of wind flows into electric energy. The instantaneous power available in a wind flow $P_V$ is defined as

$$P_V = \frac{1}{2}\rho A U_\infty^3, \tag{A-1}$$

with $A$ the cross-sectional area of the respective flow, $\rho$ the flow density and $U_\infty$ the freestream flow velocity [58]. Note that a wind turbine cannot extract all the energy from an incoming wind flow, as the flow is required to have a certain velocity behind the turbine rotor, as explained by momentum theory [11]. Momentum theory is the topic of Section 2-1-6.

Wind turbines interact with freestream flow to capture energy. The resulting flow downstream, the "flow wake", has a number of properties [12, 59]:

- Decreased flow velocity, as energy is extracted by the turbine.
- Increased turbulence, due to interactions with the turbine structure.
- A wake structure expanding with downstream distance from the turbine. According to the law of conservation of mass and assuming incompressibility of the flow, a decrease in flow velocity results in a proportional increase in the cross-sectional area of the wake.
- A skewed and deflected wake structure due to the (direction of) rotation of the turbine rotor, inducing asymmetric forces on the flow. This causes the orientation of the wake structure to change slightly [12].

- Wake meandering. The entire wake structure will show spatial and temporal oscillations, rather than maintaining one fixed position and shape. This stochastic phenomenon is a nontraditional form of turbulence, implying that certain locations in the downstream flow are irregularly affected by a wake [59].

As the wake propagates further downstream, the flow recovers to the freestream conditions by convection and diffusion, promoted by turbulent forces [12]. An idealized wake in single turbine operation is demonstrated in Figure A-1, obtained using the WindFarmSimulator (WFSim) flow model. This wind farm model is described in detail in Chapter 2. Note that this figure does not display wake skewing, wake deflection, or wake meandering effects, due to the model's simplicity.

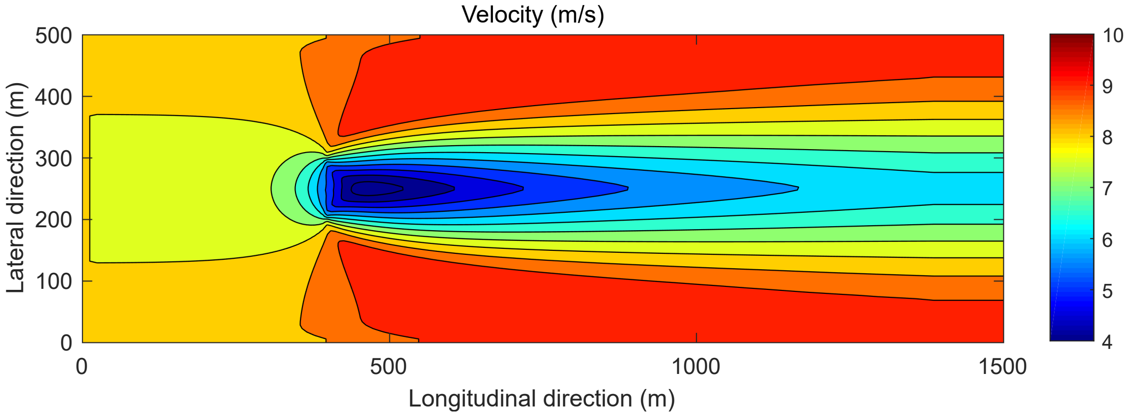


**Figure A-1:** Horizontal slice at hub height of a fully developed flow including wake in a single turbine simulation, obtained using WFSim. The turbine is positioned at $(400, 250)$ m with a rotor radius of $45$ m. The wind is flowing from left to right in the figure, with inflow $U_\infty = 8$ m/s.

Besides turbine-induced flow fluctuations, the flow velocity and direction are heavily dependent on a number of external factors, such as geographical location, climate characteristics, altitude, surface roughness and solar conditions [60]. This further complicates flow modeling and reconstruction [11]. More information on wind and wind-turbine interactions can be found in books by Tong [58] and Bianchi *et al.* [22].

## A-2 Single turbine control

Traditionally, wind turbines are controlled to maximize power capture while mitigating structural loading to acceptable levels. The applied control algorithm depends on a turbine's operating conditions. On the most basic level, three regions of operation can be distinguished, as demonstrated in Figure A-2.[1]

[1] In modern wind turbines, additional regions of operation are included for control. Common additional regions are "region 1.5" and "region 2.5", which account for problematic transitions between different control regimes. In Figure A-2, transitions between control regimes give rise to discontinuous jumps in the control signal, resulting in excessive mechanical loading [11, 12]. Furthermore, in reality, there is a fourth control region for extremely high wind speeds, at which the turbine is shut down to manage structural loading.

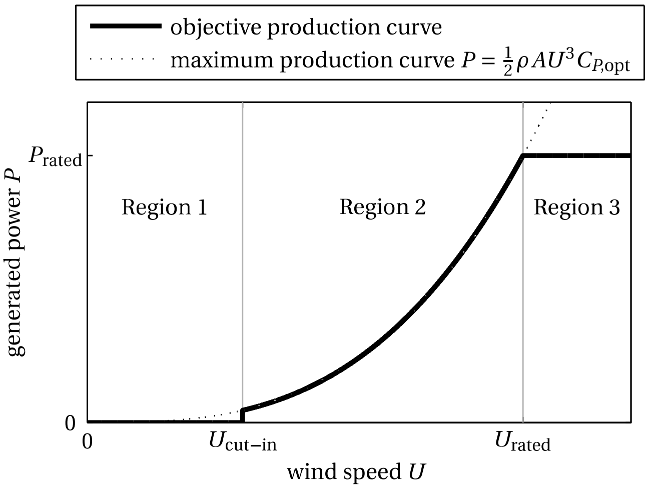


**Figure A-2:** A very basic power production curve for single turbine operation. In region 1, it is not worthwhile to operate the turbine. In region 2, the turbine is operated to capture as much power as possible. In region 3, the turbine is controlled to mitigate structural loads and maintain the turbine's power output at a rated value $P_\text{rated}$. [12]

**Region 1** In this figure, the first region depicts the (absence of a) control algorithm for very low wind speeds ($U < U_\text{cut-in}$), as turbine control is deemed not worthwhile. The turbine is not operated, and no power is generated.

**Region 2** For medium to high wind speeds ($U_\text{cut-in} \leq U < U_\text{rated}$), the turbine is operated to capture as much power $P$ from the wind as possible, according to

$$P_\text{opt} = \frac{1}{2}\rho A U^3 C_{P,\text{opt}}, \tag{A-2}$$

with $C_P$ the dimensionless power coefficient, defined as the ratio between the power captured by a turbine and the power available in the wind, by $C_P = \frac{P}{P_V}$. $C_P$ depends on the blade geometry, the pitch angles $\beta$ and the tip-speed-ratio (TSR) $\lambda$.[2] The TSR is a function of the rotational speed of the rotor $\omega$, by

$$\lambda = \frac{\omega R}{U}, \tag{A-3}$$

with $R$ the turbine rotor radius. A traditional $C_P$ curve is shown in Figure A-3.

The optimal power coefficient $C_{P,\text{opt}}$ corresponds to the TSR $\lambda_\text{opt}$ and collective pitch angle $\beta_\text{opt}$ at which maximum power is captured. $\beta$ is directly controlled using the blade pitch actuators. On the other hand, $\lambda$ is controlled indirectly through the rotor speed $\omega$, if possible.[3] $\lambda_\text{opt}$ is achieved by adjusting the generator torque $\tau_g$, as

$$\tau_{g,\text{opt}} = \frac{\rho A R^3 C_{P,\text{opt}} n}{2\lambda_\text{opt}^3}, \tag{A-4}$$

[2] For old turbines it may not be possible to pitch the blades, which renders $C_P$ as a function of $\lambda$ only.

[3] Modern, variable-speed wind turbines allow the turbine to run at different angular velocities, in contrast to fixed-speed wind turbines.

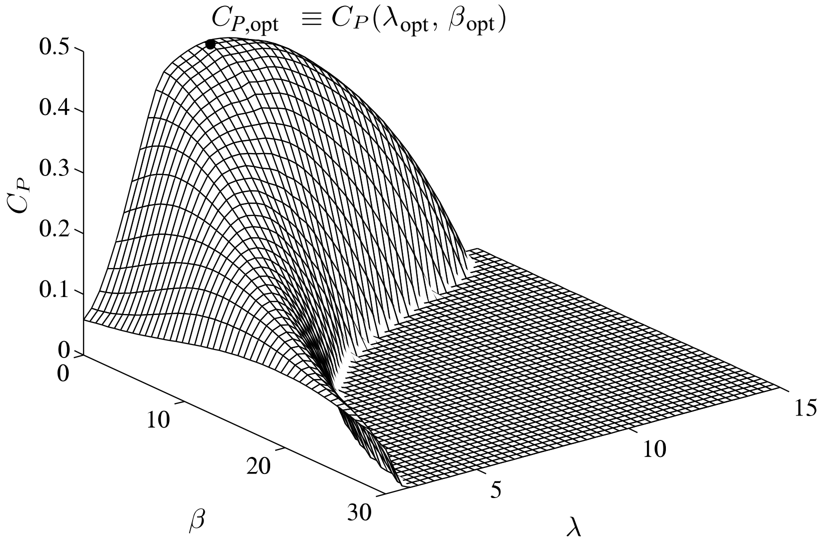


**Figure A-3:** Traditional dimensionless power coefficient curve $C_P(\lambda, \beta)$. [22]

with $n$ the gearbox ratio [12]. Furthermore, if the turbine is able to yaw, it can be aligned with the incoming wind flow to maximize the effective wind speed $U$, so that power capture can be maximized since $P \propto U^3$. As yaw actuation is relatively slow and to reduce actuator usage, a threshold is typically implemented.

**Region 3** Finally, for very high wind speeds ($U \geq U_{\text{rated}}$), maximizing power capture is detrimental for the turbine structure and power electronics. Hence, the loads are mitigated and the turbine is controlled to capture a rated power $P_{\text{rated}}$. Turbine derating is typically done by pitching the rotor blades away from $\beta_{\text{opt}}$, or yawing the turbine away from the upstream flow [11, 12], while maintaining constant generator torque $\tau_g$.

Appendix B

# Supplementary results: lateral flow velocity plots

This appendix contains the contour plots for the flow field estimates of WFSim, WFSim+ApKF, and WFSim+EnKF, respectively. These are omitted from the main text as they are less of interest for the current purposes. Namely, for control the mean wake centerline will be looked at mainly. However, fundamentally there is no significant difference between the longitudinal and lateral velocity components, and their results are expected to be conceptually similar.

## B-1 WFSim

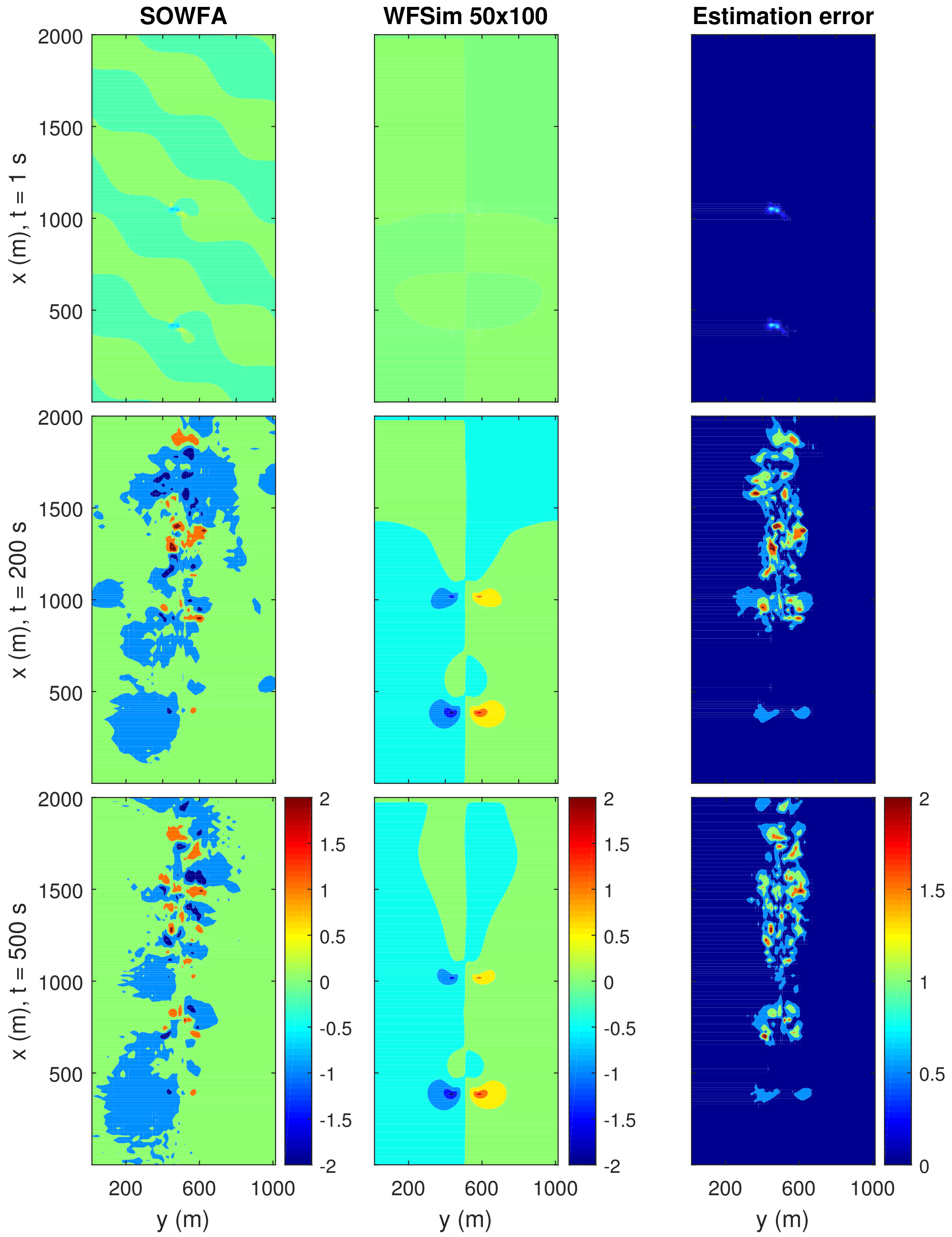
The lateral velocity plots of the validation case in Section 2-3 are displayed in Figure B-1. In these figures, it is seen that SOWFA captures more dynamics than WFSim does, as previously seen in the longitudinal velocity plots (Figure 2-10) too. For example, the simplified turbulence and turbine submodels neglect the strongly nonlinear near wake flow region. Furthermore, the main source of error is in the wake of turbine 2, as also expected from the discussion in Section 2-3-3.

## B-2 Approximate Kalman filtering

The lateral velocity plots of the ApKF observer assessment in Section 4-4 are displayed in Figure B-2. Interesting to note is that the streaky estimation error phenomenon found in the 50x100 meshing is not present in the lateral flow estimation results. Secondly, the 25x50 meshing again seems to outperform the 50x100 meshing, as also seen in the longitudinal flow plots. Furthermore, most error originates at and downstream of the second turbine.

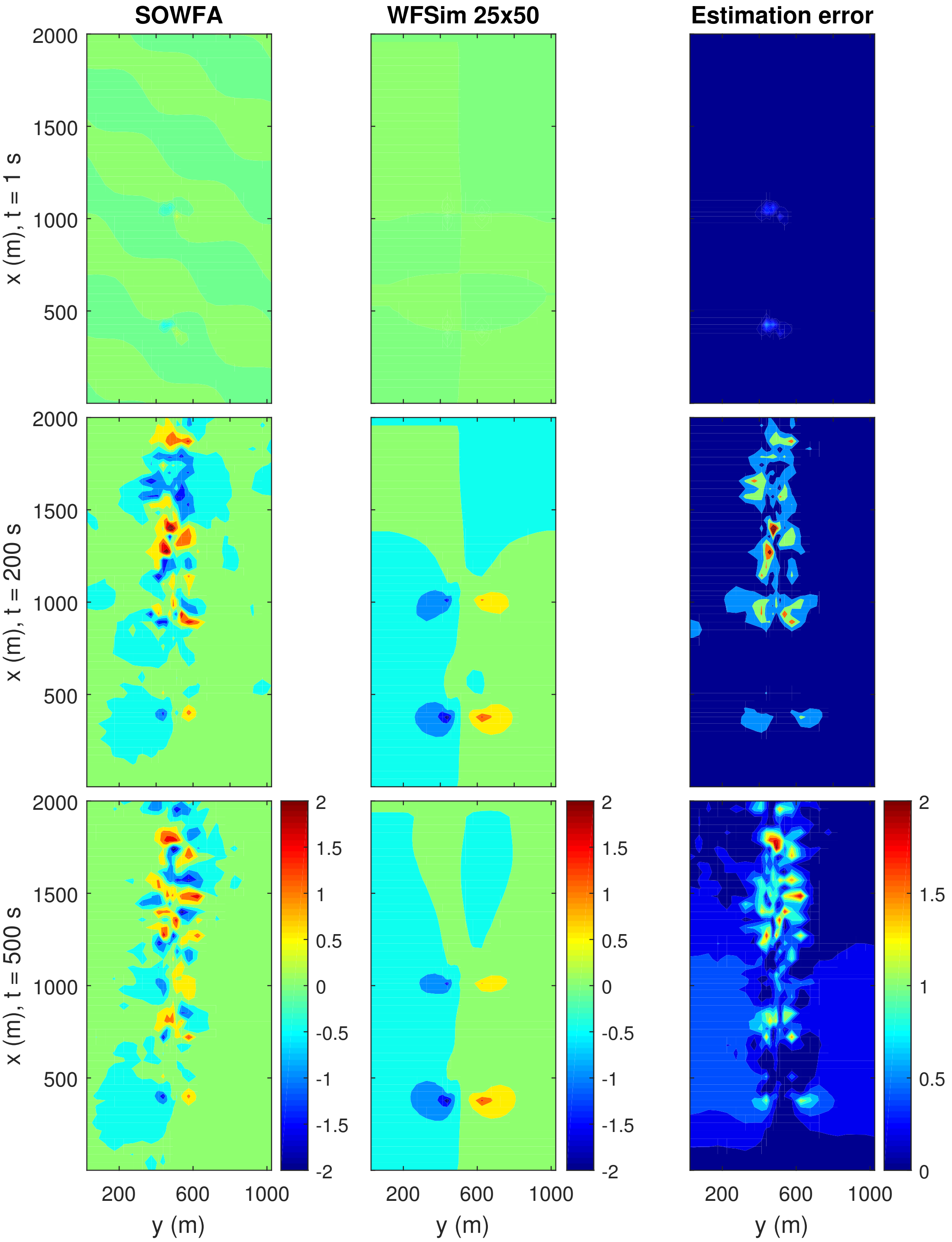
## B-3 Ensemble Kalman filtering

The lateral velocity plots of the EnKF observer assessment in Section 4-4 are displayed in Figure B-3. The flow estimates are much less uniform, similar to as seen in the longitudinal velocity plots. This is because the EnKF algorithm relies on adding artificial noise to the measurements and the model for state reconstruction. Furthermore, most estimations errors in the flow are found in the wake of turbine 2. However, the errors appear to be smaller than for the ApKF or for the case without observer, respectively Figures B-1 and B-2. This statement is consolidated by the results seen previously in Section 4-5.



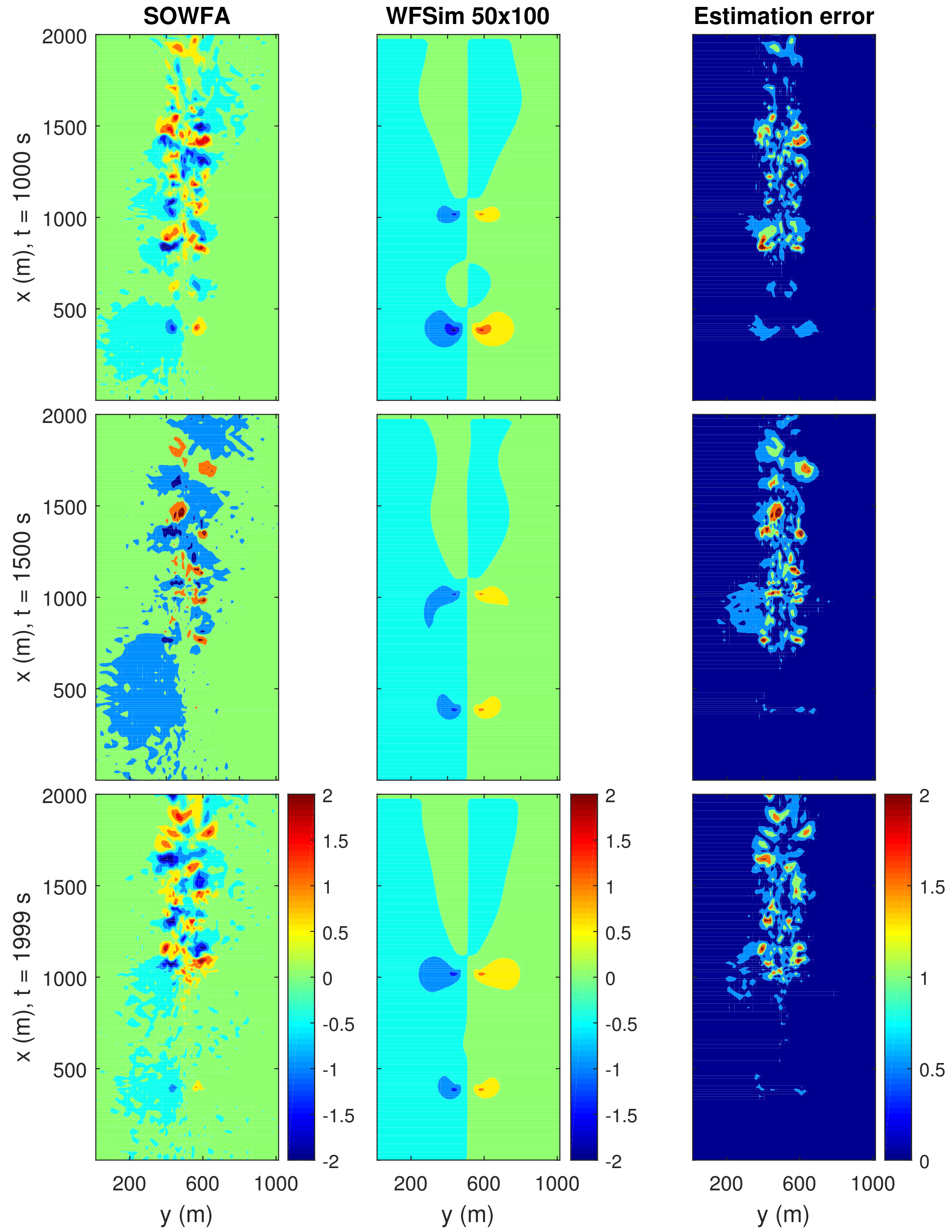
**(a)** Lateral flow velocity for 50x100 meshing at $t = 1, 200, 500$ s.

**Figure B-1:** Snapshots of the lateral flow velocity (m/s) throughout the grid for various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s for the WFSim validation case. The raw data is shown on the left using SOWFA data sampled at a 50x100 meshing. WFSim is simulated under a 50x100 and a 25x50 meshing.



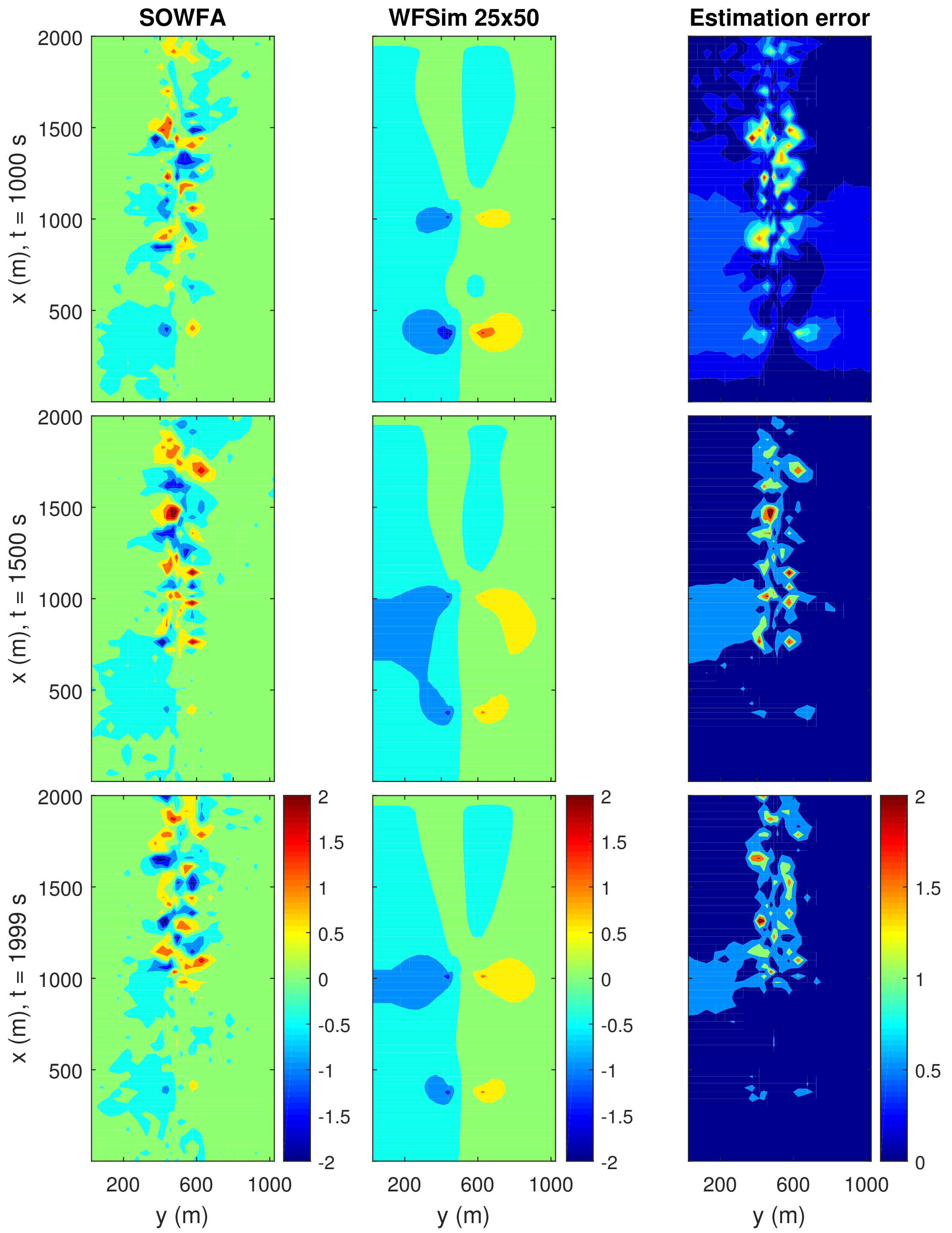
**(b)** Lateral flow velocity for 25x50 meshing at $t = 1, 200, 500$ s.

**Figure B-1:** Snapshots of the lateral flow velocity (m/s) throughout the grid for various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s for the WFSim validation case. The raw data is shown on the left using SOWFA data sampled at a 50x100 meshing. WFSim is simulated under a 50x100 and a 25x50 meshing.



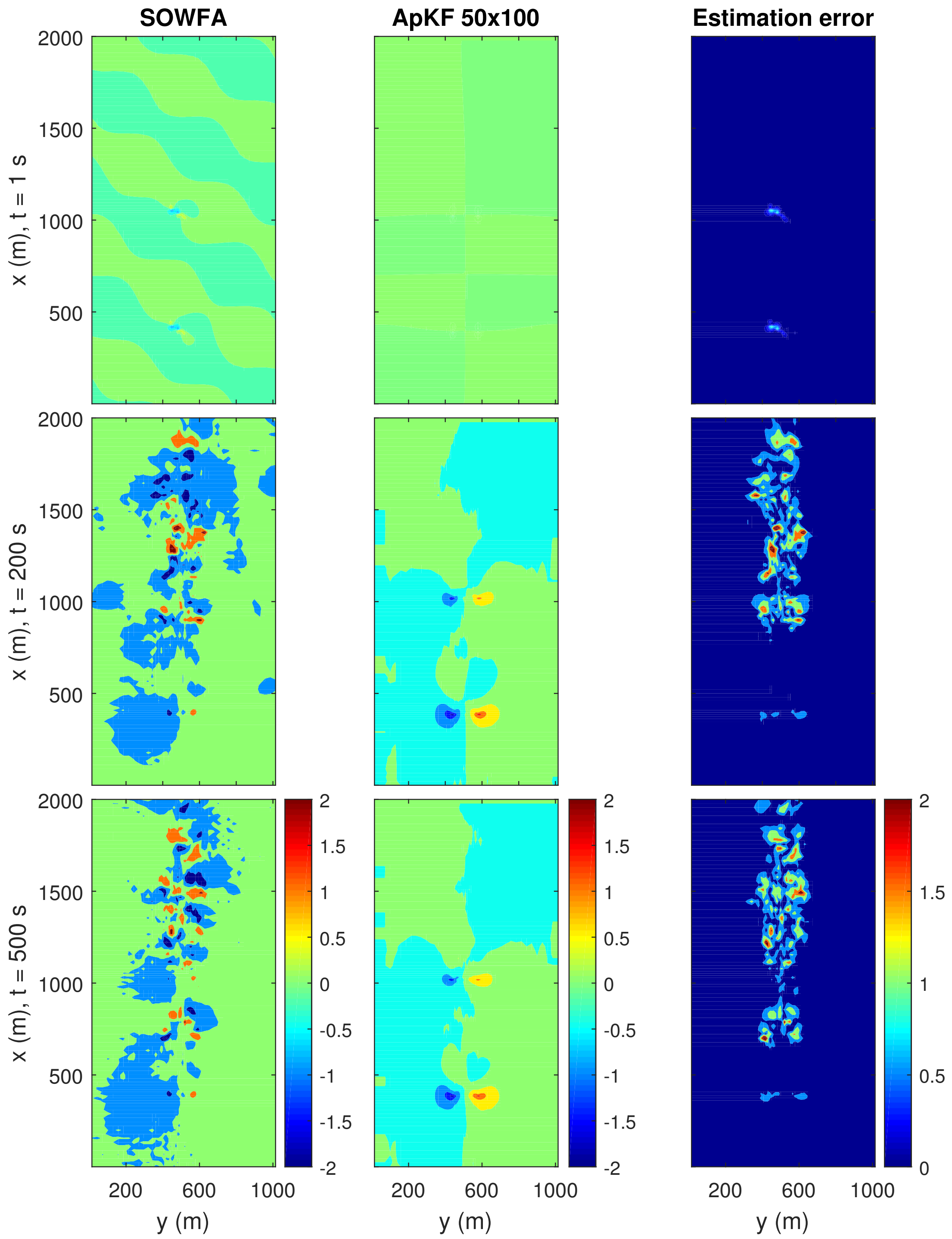
**(c)** Lateral flow velocity for 50x100 meshing at $t = 1000, 1500, 1999$ s.

**Figure B-1:** Snapshots of the lateral flow velocity (m/s) throughout the grid for various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s for the WFSim validation case. The raw data is shown on the left using SOWFA data sampled at a 50x100 meshing. WFSim is simulated under a 50x100 and a 25x50 meshing.



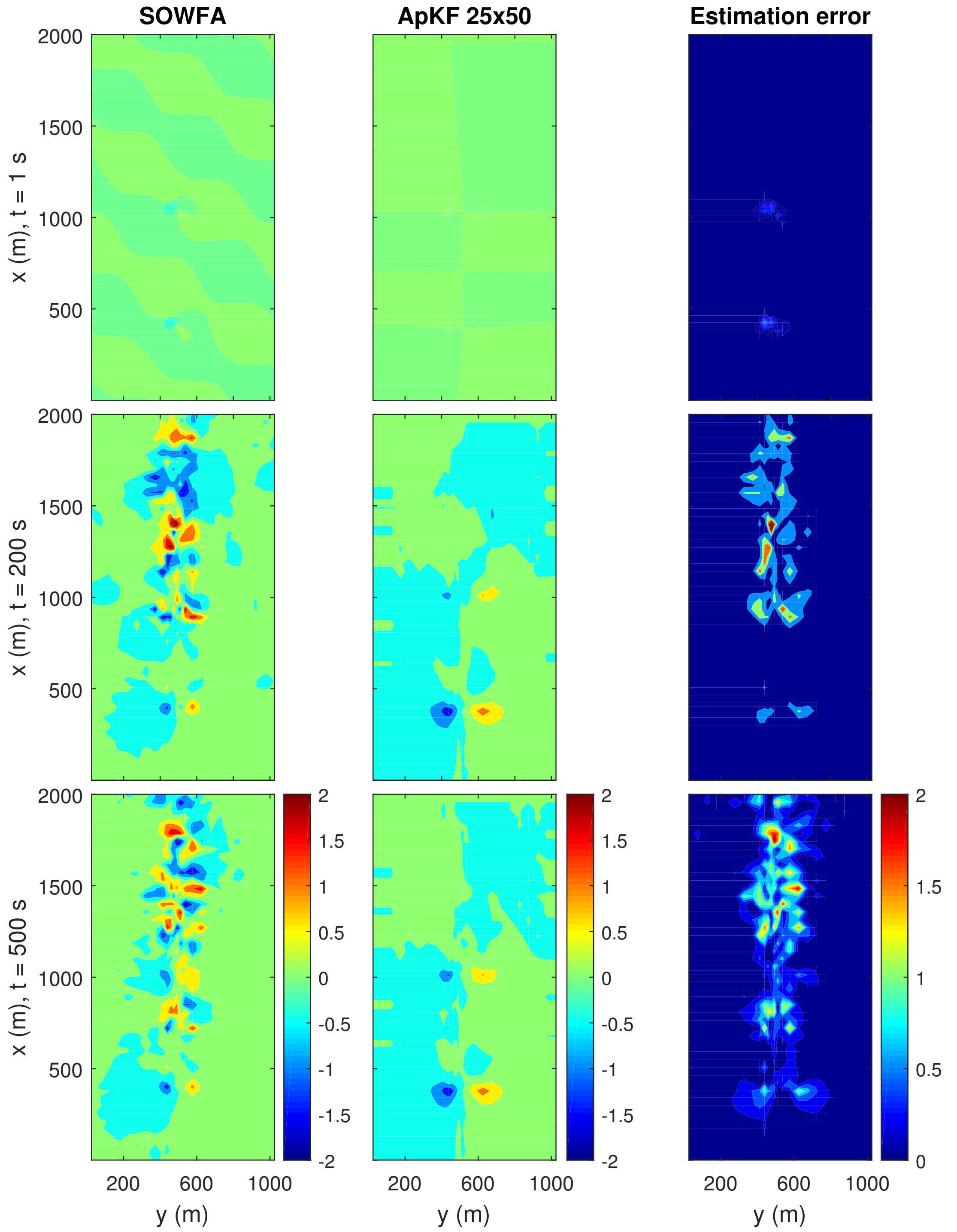
**(d)** Lateral flow velocity for 25x50 meshing at $t = 1000, 1500, 1999$ s.

**Figure B-1:** Snapshots of the lateral flow velocity (m/s) throughout the grid for various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s for the WFSim validation case. The raw data is shown on the left using SOWFA data sampled at a 50x100 meshing. WFSim is simulated under a 50x100 and a 25x50 meshing.



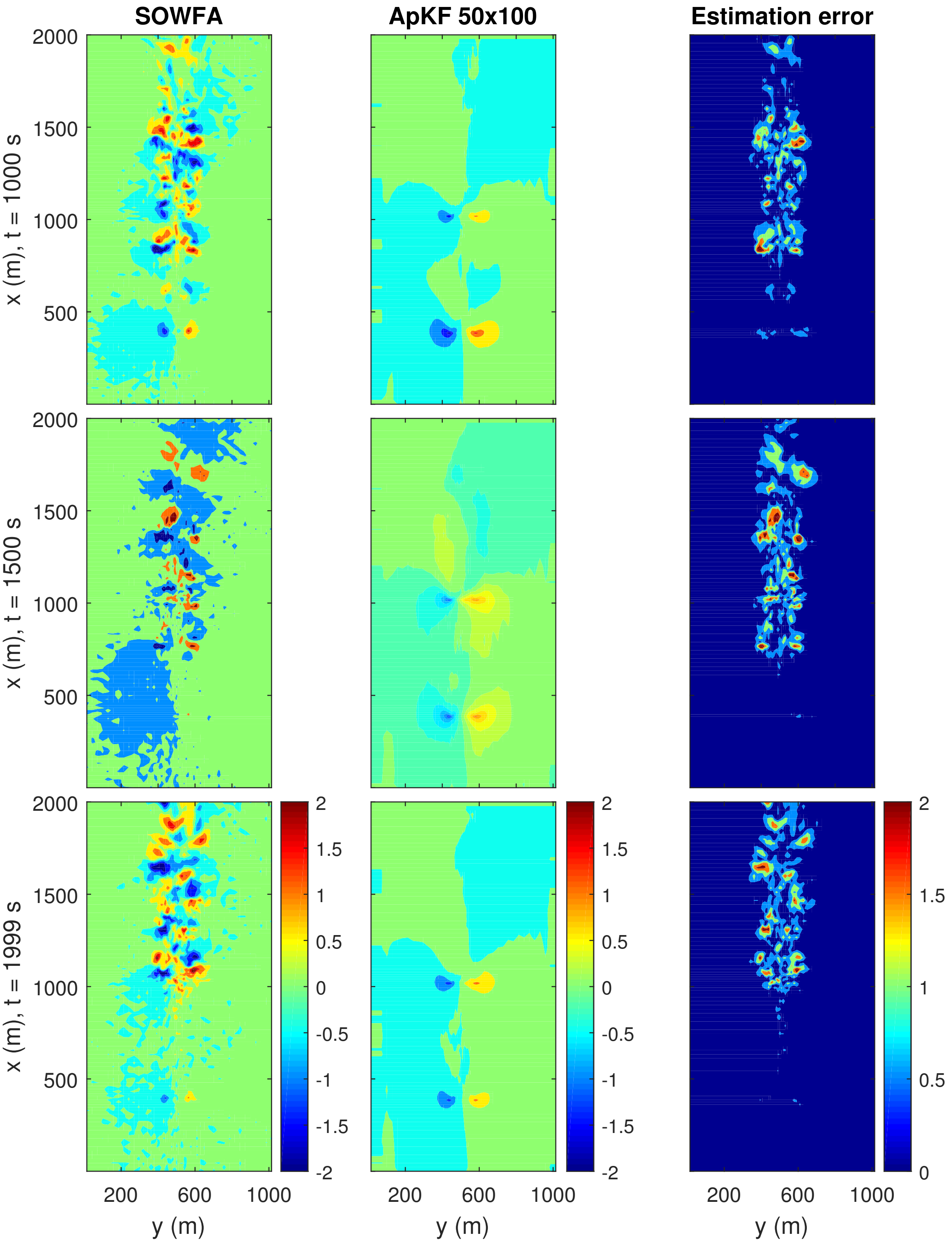
**(a)** Lateral flow velocity for 50x100 meshing at $t = 1, 200, 500$ s.

**Figure B-2:** Snapshots of the lateral flow velocity (m/s) throughout the grid for various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s for the ApKF. The raw data is shown on the left using SOWFA data sampled at a 50x100 meshing. The ApKF is simulated under a 50x100 and 25x50 meshing.



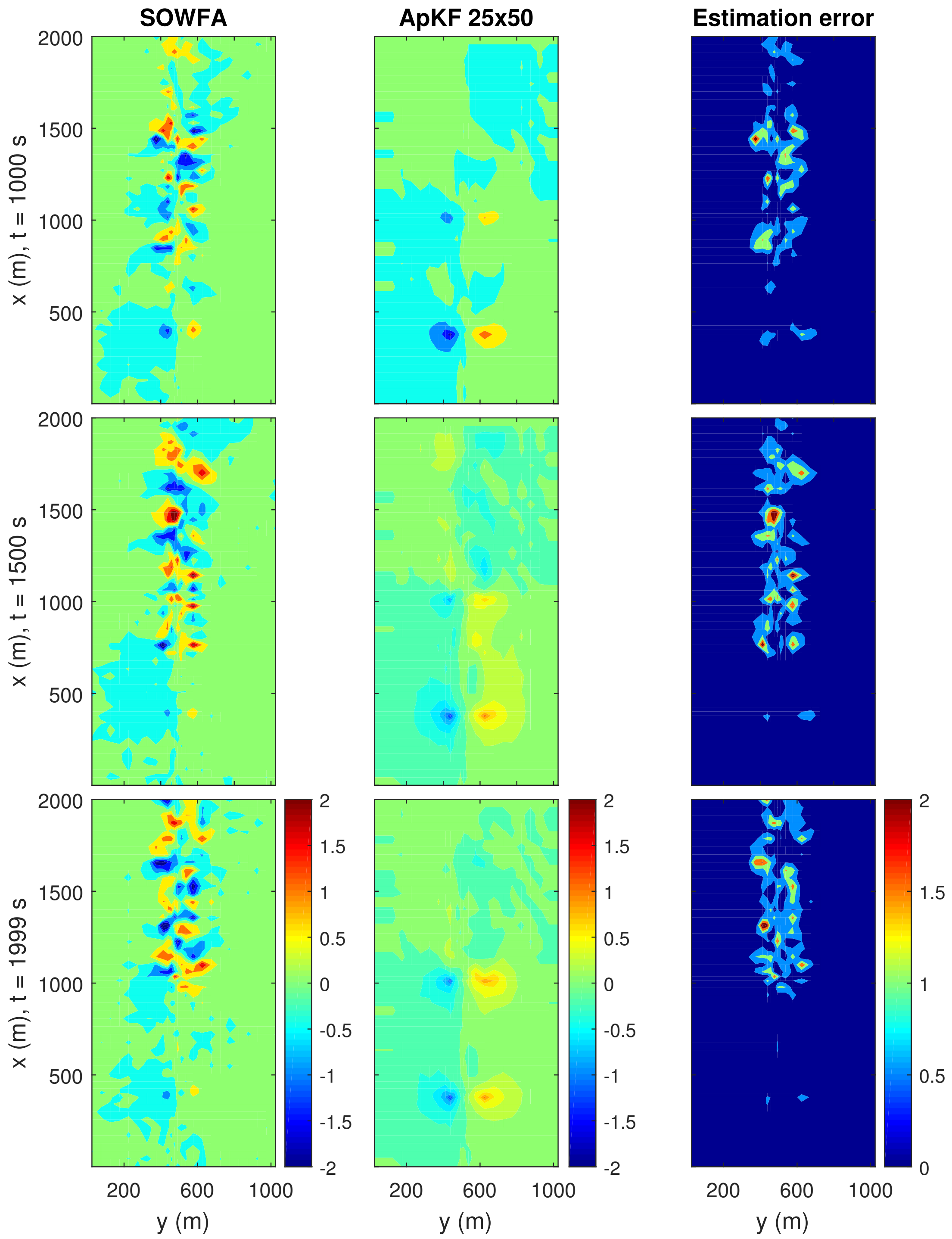
**(b)** Lateral flow velocity for 25x50 meshing at $t = 1, 200, 500$ s.

**Figure B-2:** Snapshots of the lateral flow velocity (m/s) throughout the grid for various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s for the ApKF. The raw data is shown on the left using SOWFA data sampled at a 50x100 meshing. The ApKF is simulated under a 50x100 and 25x50 meshing.



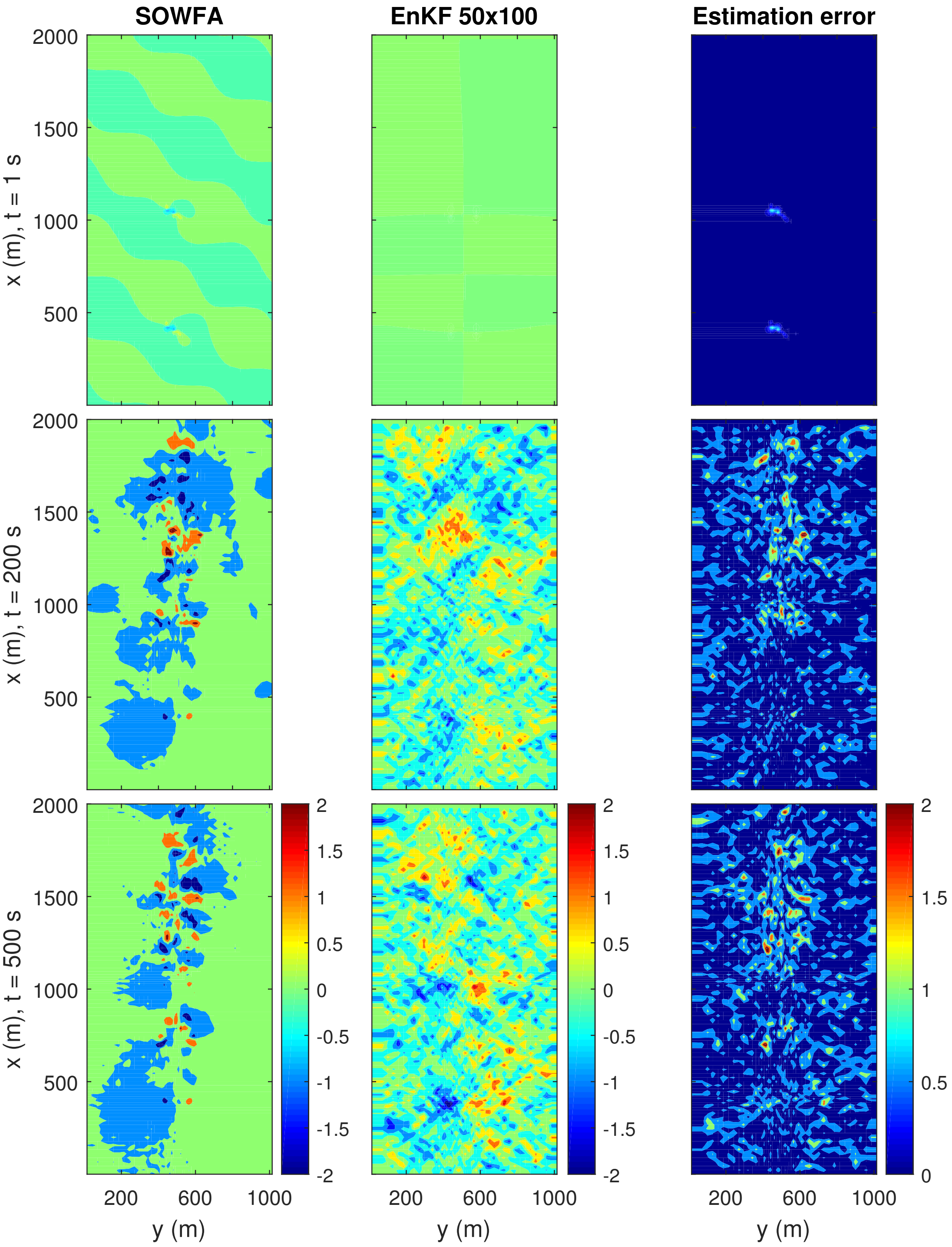
**(c)** Lateral flow velocity for 50x100 meshing at $t = 1000, 1500, 1999$ s.

**Figure B-2:** Snapshots of the lateral flow velocity (m/s) throughout the grid for various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s for the ApKF. The raw data is shown on the left using SOWFA data sampled at a 50x100 meshing. The ApKF is simulated under a 50x100 and 25x50 meshing.



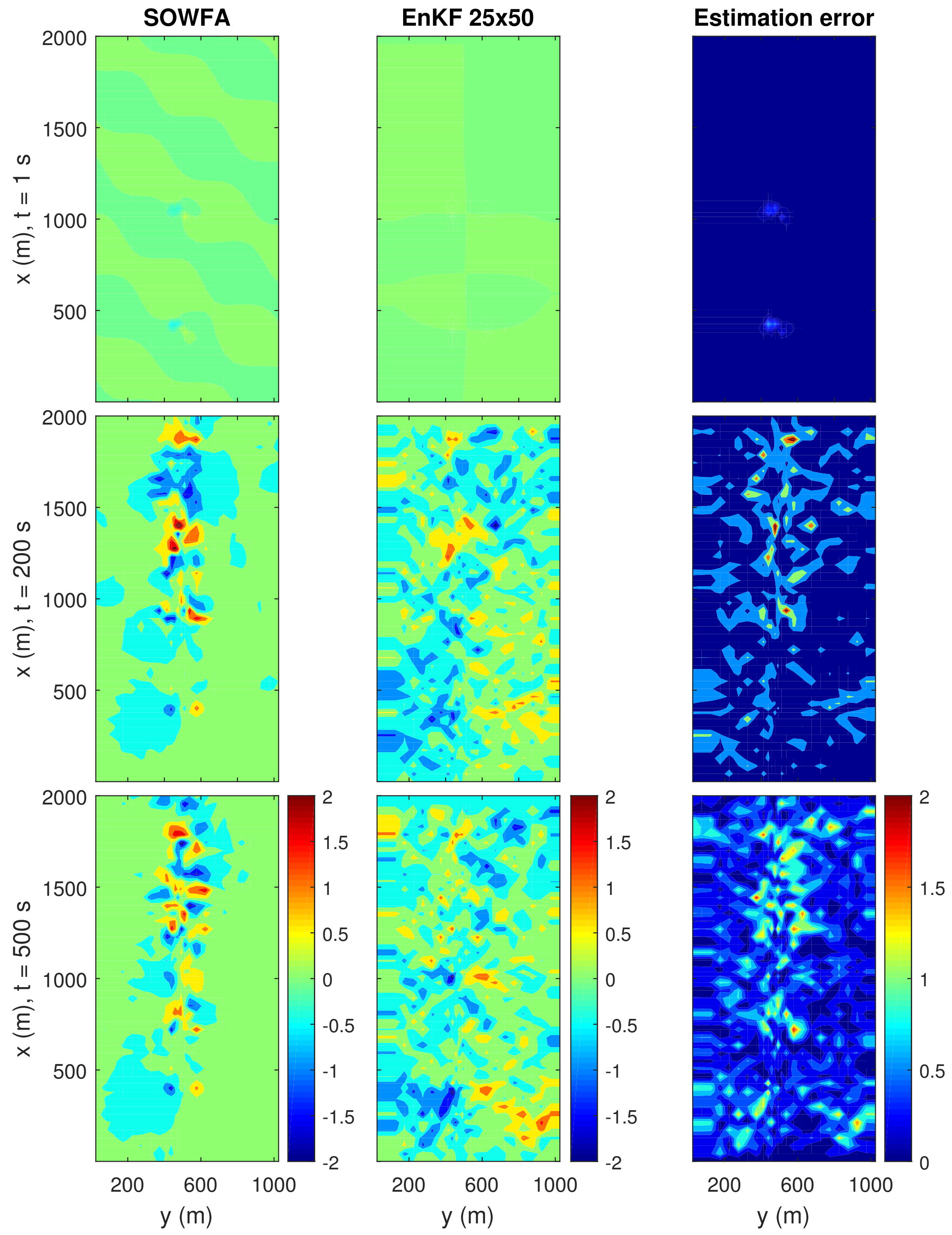
**(d)** Lateral flow velocity for 25x50 meshing at $t = 1000, 1500, 1999$ s.

**Figure B-2:** Snapshots of the lateral flow velocity (m/s) throughout the grid for various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s for the ApKF. The raw data is shown on the left using SOWFA data sampled at a 50x100 meshing. The ApKF is simulated under a 50x100 and 25x50 meshing.



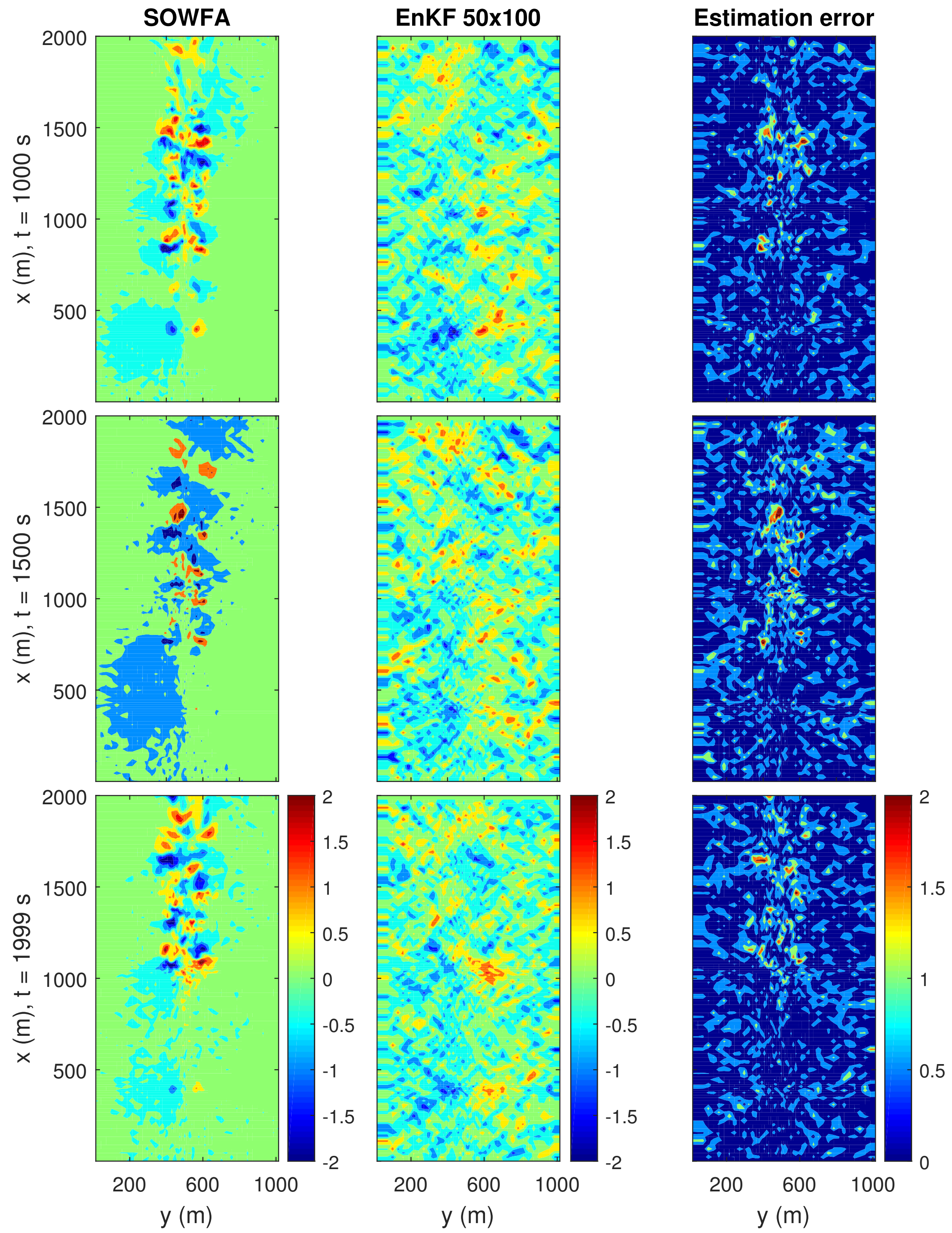
**(a)** Lateral flow velocity for 50x100 meshing at $t = 1, 200, 500$ s.

**Figure B-3:** Snapshots of the lateral flow velocity (m/s) throughout the grid for various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s for the EnKF. The raw data is shown on the left using SOWFA data sampled at a 50x100 meshing. The EnKF is simulated under a 50x100 and 25x50 meshing.



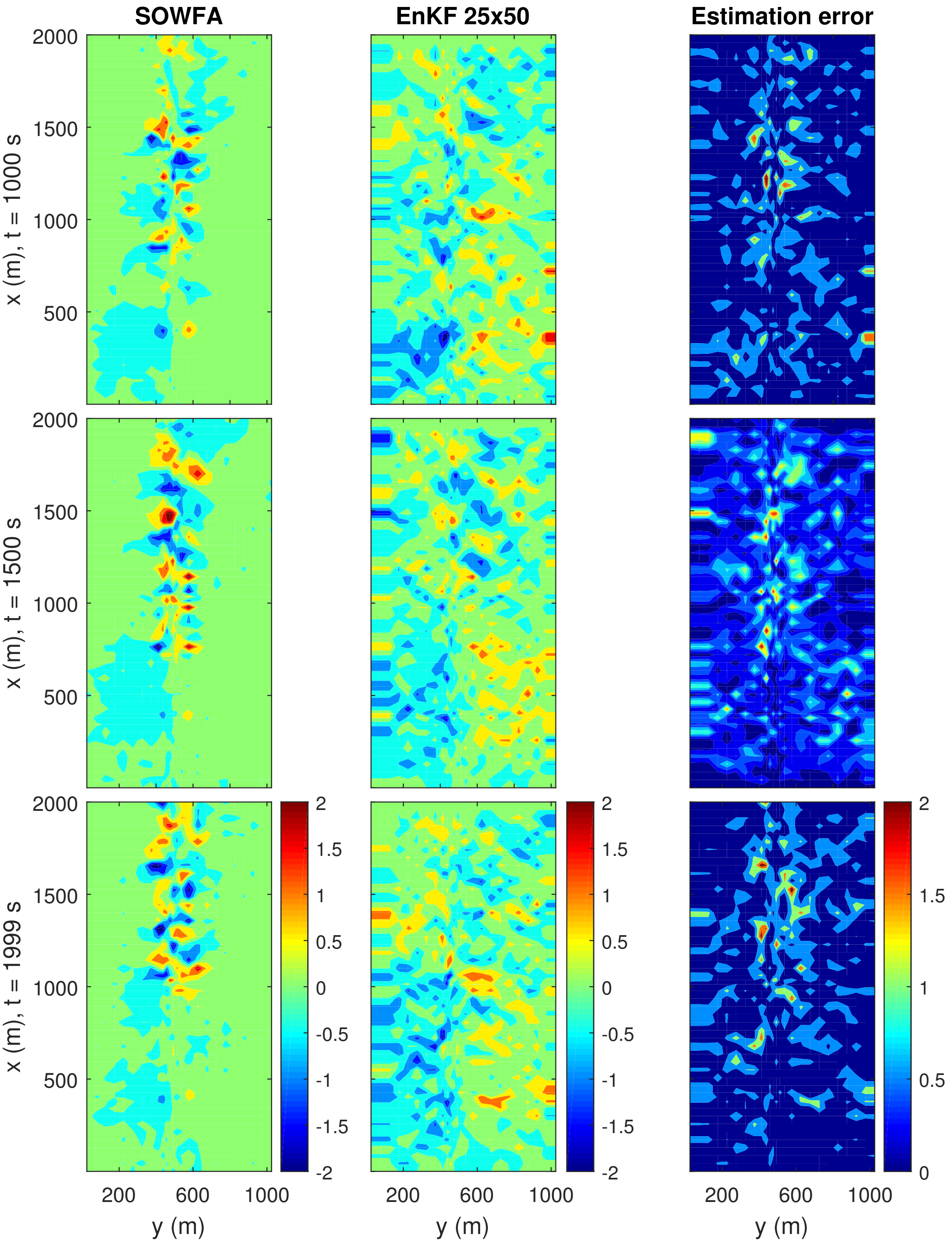
**(b)** Lateral flow velocity for 25x50 meshing at $t = 1, 200, 500$ s.

**Figure B-3:** Snapshots of the lateral flow velocity (m/s) throughout the grid for various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s for the EnKF. The raw data is shown on the left using SOWFA data sampled at a 50x100 meshing. The EnKF is simulated under a 50x100 and 25x50 meshing.



**(c)** Lateral flow velocity for 50x100 meshing at $t = 1000, 1500, 1999$ s.

**Figure B-3:** Snapshots of the lateral flow velocity (m/s) throughout the grid for various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s for the EnKF. The raw data is shown on the left using SOWFA data sampled at a 50x100 meshing. The EnKF is simulated under a 50x100 and 25x50 meshing.



**(d)** Lateral flow velocity for 25x50 meshing at $t = 1000, 1500, 1999$ s.

**Figure B-3:** Snapshots of the lateral flow velocity (m/s) throughout the grid for various time instants $t = 1, 200, 500, 1000, 1500, 1999$ s for the EnKF. The raw data is shown on the left using SOWFA data sampled at a 50x100 meshing. The EnKF is simulated under a 50x100 and 25x50 meshing.

# Appendix C

# Paper submission for TORQUE

The work presented in this thesis has resulted in a paper submission to the Science of Making Torque from Wind (TORQUE) conference in Munich, Germany. This conference will take place in early October, 2016. This submission has been accepted for a poster presentation. The next ten pages show the most recent draft version of this paper submission, written in addition to this dissertation.

Please note that this document has its own list of references, acronyms, and footnotes. This paper is readable as a stand-alone document, meant to summarize the main contributions to literature by this master's thesis.

# Enhanced Kalman Filtering for a 2D CFD NS Wind Farm Flow Model

**B M Doekemeijer**[1], **J W van Wingerden**[1], **S Boersma**[1] **and L Y Pao**[2]

[1] Faculty of Mechanical Engineering, Delft University of Technology, The Netherlands
[2] Faculty of Electrical Engineering, University of Colorado Boulder, United States

E-mail: bart@doekemeijer.nl

**Abstract.** Wind turbines are often grouped together for financial reasons, but due to wake development this usually results in decreased turbine lifetimes and power capture, and thereby an increased levelized cost of energy (LCOE). Wind farm control aims to minimize this cost by operating turbines at their optimal control settings. Most state-of-the-art control algorithms are open-loop and rely on a low fidelity, static flow model. Closed-loop control relying on a dynamic model and state observer has real potential to further decrease wind's LCOE, but is often too computationally expensive for practical use. In this paper two time-efficient Kalman filter (KF) variants are outlined incorporating the medium fidelity, dynamic flow model "WindFarmSimulator" (WFSim). This model relies on a discretized set of Navier-Stokes equations in 2D to predict the flow in wind farms at hub height at low computational cost. The filters implemented are an Ensemble KF and an Approximate KF. Simulations in which a high fidelity simulation model represents the true wind farm show that these filters are typically $10^1 - 10^2$ times faster than a regular KF with comparable or better performance, correcting for wake dynamics that are not modeled in WFSim (noticeably, wake meandering and turbine hub effects). This is a first big step towards real-time closed-loop control in wind farms.

## 1. Introduction

The recent 2015 UN Climate Change Conference in Paris once again stressed the importance of renewable energy sources, among which is wind. To make implementations of wind energy feasible, its levelized cost of energy (LCOE)[1] has to be able to compete with that of other sources. For this reason, wind turbines are often placed together in wind farms. However, grouping turbines together gives rise to the development of, and interactions with, wind wakes, often resulting in a plant-wide decreased power capture[2] and increased turbine structural loading compared to an equal number of single turbines. Wind farm control aims to counter this and minimize the LCOE of wind. Notable findings in this relatively new area of research include farm layout optimization [1, 2], wake redirection control [2, 3] and active power control [4, 5].

Advancements in wind farm control have gone hand in hand with advancements in modeling, as state-of-the-art control algorithms typically rely on a low fidelity, control-oriented, static flow model. Unfortunately, this is often limited to open-loop control. Furthermore, high fidelity, dynamic models based on the Navier-Stokes (NS) equations have allowed relatively inexpensive flow analyses and observer testing at high accuracy, but are computationally too expensive for

[1] The LCOE is a measure to compare different methods of energy generation in terms of financial feasibility.

[2] Power capture losses in wind plants may be up to 50%, according to some numerical studies [1].

real-time control applications. Medium fidelity, dynamic flow models aim to bridge this gap, and in some cases allow closed-loop control, further reducing the LCOE of wind. However, the need for a state observer and the increase in computational cost remain challenges.

The goal of this work is to design a time-efficient state observer using the medium fidelity, dynamic flow model "WindFarmSimulator" (WFSim), in the pursuit of real-time closed-loop control. Observers will provide estimations of the flow field correcting for unmodeled dynamics (thus: allowing the use of a lower fidelity, time-efficient model), measurement noise and limited availability (temporally and spatially) of measurements. The concept is visualized in figure 1.

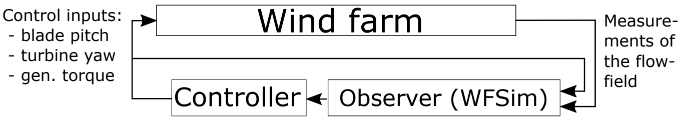


Figure 1: Closed-loop control for wind farms.

In this paper, first the WFSim model will be outlined in section 2. Secondly, two enhanced KF designs will be outlined in section 3. Thirdly, the simulation setup and simulation results will be described in sections 4 and 5, respectively. Finally, conclusions are drawn in section 6.

## 2. WindFarmSimulator

This section first introduces WFSim, after which modifications are described to allow direct implementation in state-of-the-art filtering algorithms. For a more detailed explanation of these equations, please see Doekemeijer's dissertation [6].

### *2.1. The flow model*

WFSim is a medium fidelity, dynamic flow model from the Delft University of Technology [7, 8]. It predicts the flow velocity vectors in a wind farm at hub height in predefined meshings using the spatially and temporally discretized 2D NS equations following a computational fluid dynamics (CFD) solution [9]. WFSim employs the actuator disk model (ADM) [10] to calculate the aerodynamic forces on the flow by the rotor blades, approximating the turbine rotor by an infinitely thin disk uniformly extracting energy from the flow. Furthermore, it includes a mixing length wake turbulence model [11] to account for wake recovery, valid for slow changes in the flow direction. What makes WFSim unique is its reformulation into an implicit, nonlinear state space system by projecting away the continuity equations, resulting in the flow model

$$\begin{aligned} E(\alpha_k)\alpha_{k+1} =& A\ \alpha_k + B(\alpha_k)\beta_k + S(\alpha_k), \\ \begin{bmatrix} \overline{u}_k \\ \overline{v}_k \end{bmatrix} =& Q_p\alpha_k + B_p \end{aligned} \tag{1}$$

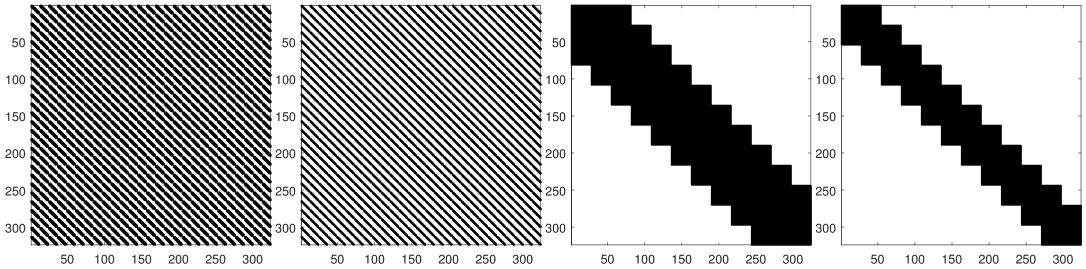
with $\alpha_k \in \mathbb{R}^N$ the system state vector at time $k$, and $\left[\overline{u}_k; \overline{v}_k\right] \in \mathbb{R}^M$ the collocated 2D velocity vectors in the grid. $N$ is proportional to the number of grid points in the mesh (i.e. refinement of flow fields). The system input is $\beta_k \in \mathbb{R}^O$, with $O$ equal to the number of turbines. $\beta_k$ is a scaled measure of the axial induction factors, and can be translated into physical turbine settings such as generator torque, blade pitch and yaw. $A \in \mathbb{R}^{N\times N}$, $B_p \in \mathbb{R}^M$, $B \in \mathbb{R}^{N\times O}$, $E \in \mathbb{R}^{N\times N}$, $Q_p \in \mathbb{R}^{M\times N}$ and $S \in \mathbb{R}^N$ are system matrices, of which $E$, $B$ and $S$ depend on state $\alpha_k$.

### *2.2. Model manipulation*

While literature on state observers for implicit systems such as equation (1) exists (e.g., [12, 13]), for WFSim these algorithms are too computationally heavy for real-time control and often lead to numerical instability. Hence, a different approach is followed. First, equation (1) is reformulated in a more common form by extending the state vector with a constant entry allowing elimination of the offset terms $S(\alpha_k)$ and $B_p$, and performing matrix inversion of $E$, resulting in

$$\begin{aligned} \begin{bmatrix} \alpha_{k+1} \\ 1 \end{bmatrix} &= \overbrace{\begin{bmatrix} E(\alpha_k)^{-1}A & E(\alpha_k)^{-1}S(\alpha_k) \\ 0 & 1 \end{bmatrix}}^{F(\alpha_k)} \begin{bmatrix} \alpha_k \\ 1 \end{bmatrix} + \begin{bmatrix} E(\alpha_k)^{-1}B(\alpha_k) \\ 0 \end{bmatrix} \beta_k, \\ \begin{bmatrix} \overline{u}_k \\ \overline{v}_k \end{bmatrix} &= \begin{bmatrix} Q_p & B_p \end{bmatrix} \begin{bmatrix} \alpha_k \\ 1 \end{bmatrix}. \end{aligned} \tag{2}$$

In its current form, the calculation of $E^{-1}A$ at each time instant is too computationally heavy for control applications due to high dimensionality (typically, $N = 10^2 - 10^3$). Therefore, computational cost is reduced by applying the Reverse Cuthill-McKee (RCM) algorithm on equation (2), typically resulting in a computational effort reduction of 90% or more [14, 15]. The RCM algorithm rearranges the rows and columns of a matrix to transform a sparse matrix into a sparse banded matrix, as demonstrated in figure 2.



(a) E $\in \mathbb{R}^{324\times324}$ (b) A$\in \mathbb{R}^{324\times324}$ (c) RCM(E) $\in \mathbb{R}^{324\times324}$ (d) RCM(A) $\in \mathbb{R}^{324\times324}$

Figure 2: System matrices $E$ and $A$ under the RCM algorithm, in which respectively 63% and 76% of the entries are zero. This typically reduces computational cost of $E^{-1}A$ by over 90%.

The system in equation (2) allows direct implementation of state-of-the-art estimators. Note that high dimensionality is still present, and thus emphasis is put on time-efficient algorithms.

## 3. Observer design

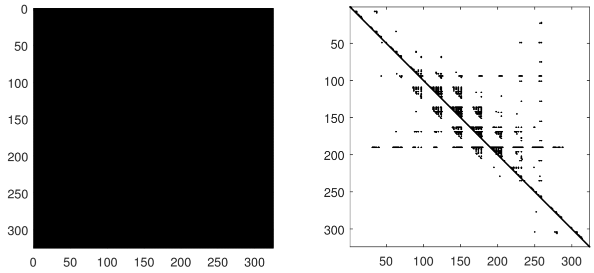
Two observers are implemented, both derived from the traditional Kalman filter (KF). However, they are fundamentally different in their way of reducing computational cost. The first filter, the Approximate KF (ApKF), simplifies the system model while retaining the original update equations. The second filter, the Ensemble KF (EnKF), instead relies on the original system model, yet approximates the KF update algorithm. Both algorithms are shortly described next.

### *3.1. Approximate Kalman filter*

The ApKF relies on the original KF update algorithm, while enforcing sparsification of two main matrices in pursuit of reducing computational cost. Firstly, all off-diagonal elements in the state covariance matrix $P_k$ are neglected, implicitly assuming the system states are uncorrelated. From numerical results it is noted that the diagonal elements are typically a factor $10^1 - 10^2$ larger than the off-diagonal elements, validating this assumption. Secondly, $F(\alpha_k)$ in equation (2) is made sparse by neglecting all matrix entries $[i, j]$ that meet the condition

$$|F(\alpha_0)[i,j] \cdot \alpha_0(j)| < \text{mean}\left(|F(\alpha_0)[i,:] \cdot \alpha_0|\right) \cdot z, \tag{3}$$

with $z$ typically $0.05 - 0.10$, and $i$ and $j$ the row and column of the matrix, respectively. For $z = 0$, the original $F$ matrix is retrieved. This algorithm enforces a sparsification by neglecting matrix entries that only have a small contribution to the system update, thereby leaving the resulting update approximately intact. This sparsification typically renders over 95% of the entries sparse, as demonstrated in figure 3 for system matrix $F(\alpha_k) \in \mathbb{R}^{324\times 324}$ with $z = 0.05$.



(a) $F \in \mathbb{R}^{324\times 324}$ (b) $F_{\text{sparsified}} \in \mathbb{R}^{324\times 324}$

Figure 3: System matrix $F$ under sparsification according to equation (3), with $z = 0.05$. This typically reduces computational cost of the filter by over 95%.

Furthermore, for $z < 0.10$, typically no loss in accuracy is noted in simulation. An example comparing the ApKF with the KF is displayed in figure 4. These simplifications reduce computation time by a factor $10^1 - 10^2$ at negligible loss in state reconstruction accuracy.

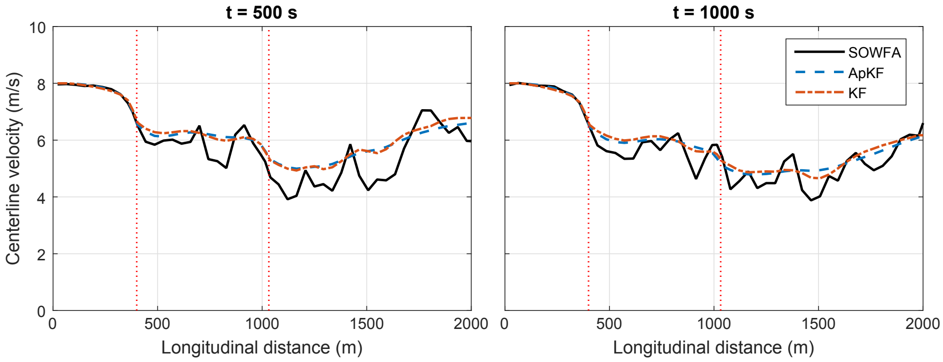


Figure 4: Simulation results showing the mean wake centerline for the ApKF ($z = 0.05$) and the KF. See Section 5 for definitions and an explanation of the simulation setup.

### *3.2. Ensemble Kalman filter*

The second observer is an EnKF. It is a suboptimal variant of the KF in which the covariance matrices are approximated by a sample covariance. Each ensemble member is a hypothesized state vector. The EnKF is typically applied to nonlinear models of high order ($10^3 - 10^6$ states) for its computational efficiency and ability to deal with nonlinear dynamics. More information concerning the EnKF can be found in a paper by Evensen [16]. The algorithm from the same paper is implemented for its emphasis on computational performance.[3]

[3] Note that there is an important difference between Evensen's algorithm and the algorithm used in WFSim. In Evensen's algorithm, at each time step, the system matrices have to be recalculated for each ensemble member due to the dependency of the system matrices on the states. With a typical ensemble size of $10^2$ members, this is very time consuming. Therefore, here, at each time instant the mean of all ensemble members is used to determine one set of system matrices, which are then used to update all ensemble members forward in time.

## 4. Simulation setup

The observers discussed in section 3 will be tested using the high fidelity simulation model SOWFA in section 5, First, in section 4.1 SOWFA will be discussed, after which the simulation domain is depicted in section 4.2. Finally, system inputs and atmospheric properties are mapped from the SOWFA simulation to WFSim, presented in section 4.3.

### *4.1. SOWFA*

Simulator fOr Wind Farm Applications (SOWFA) is a simulation model of the National Renewable Energy Laboratory (NREL) that provides highly accurate flow data at a fraction of the cost of field tests. SOWFA predicts the 3D velocity vectors in a CFD formulation using a large-eddy simulation (LES) method. It relies on the 3D incompressible NS equations for a steady or unsteady flow field, accounting for bouyancy (based on Boussinesq approximation [17]) and Coriolis effects [18]. LES methods resolve larger scale dynamics directly, but employ a subgrid-scale model for small eddy dynamics to reduce computational cost.

For rotor modeling SOWFA employs a more sophisticated version of the ADM: the actuator line model (ALM). Unlike the ADM, the ALM includes individual rotor blade effects [17].

The FAST model [19] is implemented for turbine modeling. This model calculates, among others: the power production of the turbine, the blade forces on the flow, the structural loading on the turbine, and the dynamics of several turbine components [3].

SOWFA has been used on multiple occasions for model validation, controller testing, and flow analysis in wind farms (e.g., in [2, 3, 17, 20, 21]). Its validation is still an ongoing process. Currently, field tests have shown accurate simulation results for the first 5 rows of turbines [21]. For a full description of the flow equations, please see the article by Churchfield *et al.* [22].

### *4.2. Domain and meshing*

The SOWFA dataset used for model validation and controller testing initializes simulation from a fully uniform flow field, in which no turbulence has yet developed. A two turbine[4] case with $5D$ spacing is simulated in a $3000 \times 3000 \times 1000$ m domain, with an increasingly refined meshing near the turbines and wakes. A horizontal $2000 \times 1000$ m plane at hub height is extracted from the SOWFA data as the area of interest, as displayed in figure 5.

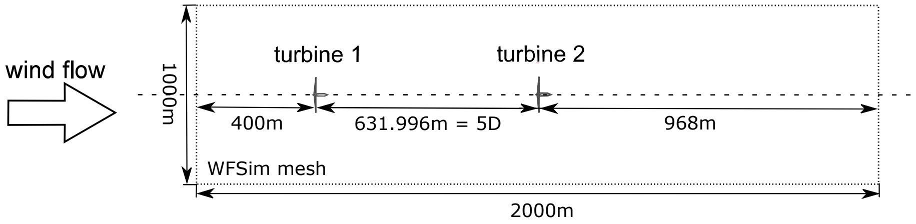


Figure 5: Simulation domain used for observer testing. [3]

In WFSim, an exponential 25x50 grid meshing is used with finer spacing near turbines and coarser spacing near the boundaries, resulting in a total number of $N = 1034$ states.

### *4.3. Mapping SOWFA to WFSim*

With the simulation domain mapped in section 4.2, now the system inputs and atmospheric settings are mapped from SOWFA to WFSim. Notably, SOWFA relies on physical turbine settings such as yaw, blade pitch angles and generator torque to determine the force turbines

[4] The NREL 5-MW turbine is used. Properties can be found in its corresponding technical report [23].

exert on the flow. WFSim on the other hand relies on the yaw angle and theoretical axial induction factor derived from momentum theory. A mapping between the two is required.

In SOWFA, the upstream turbine excites the flow by following a PRBS signal on the collective pitch angle, switching between 0° and 4°. The generator torque and yaw angle are kept constant. Turbine 2 is operated at one operating point throughout the entire simulation. The axial inductions factors are mapped accordingly. Using the tower fore-aft bending moment obtained from FAST in combination with momentum theory, the mapping is found to be as in figure 6.[5]

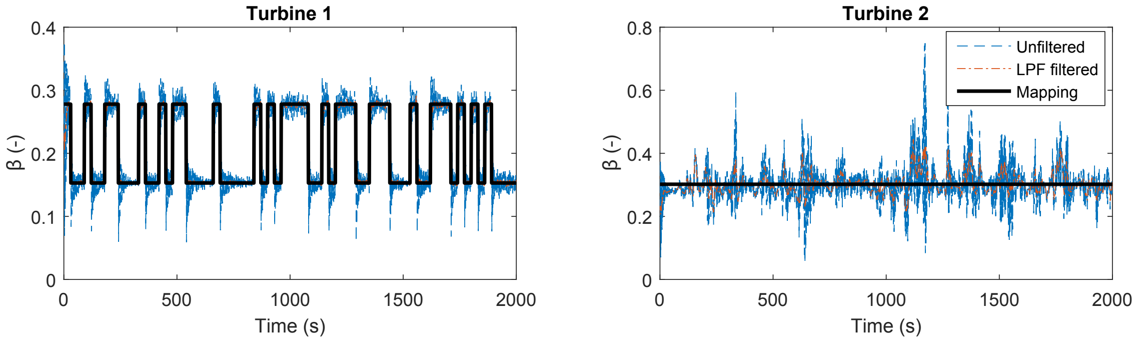


Figure 6: Estimated system input signal $\beta_k$ in equation (2).

Note that the fore-aft tower bending moment data is low-pass filtered (LPF) to remove tower vibration effects in the estimate of $\beta_k$. From this figure, it is noted that the estimate for turbine 1 is significantly better than for turbine 2, expected to be due to wake formation. Momentum theory assumes a uniform inflow in front of the turbines. For turbine 1, this is more or less the case as the simulations are initialized from a fully uniform flow field. For turbine 2, this assumption does not hold due to wake formation by the upstream turbine. This is expected to reflect in simulation results, with poorer estimations in the wake of turbine 2.

Finally, several atmospheric properties are mapped, as displayed in table 1.

Table 1: Atmospheric and model parameters for WFSim.

| Parameter [units] | Symbol | Mapping (WFSim) | True value (SOWFA) |
|---|---|---|---|
| Air density [kg·m$^3$] | $\rho$ | 1.2231 | N.A. (Time-varying) |
| Viscosity [Pa·s] | $\mu$ | $0.18 \cdot 10^{-5}$ | N.A. |
| Free-stream flow speed (long.) [m/s] | $U_\infty$ | 8.00 | $8.00 \pm 0.14$ |
| Free-stream flow speed (lat.) [m/s] | $V_\infty$ | 0.00 | $0.02 \pm 0.04$ |
| Mixing length for turbulence model [m] | $l_m$ | 0.60 | N.A. |

Using the simulation settings as outlined in this section, the WFSim farm model can be validated with SOWFA, and the ApKF and the EnKF can be tested and compared.

## 5. Results

Simulations are performed at a 25x50 grid meshing with $N = 1034$ system states. Equidistantly spaced measurements and measurements around the turbines are fed into the observer, resulting in 23% of the system outputs available to the ApKF and EnKF. These measurements are

[5] For details, see Doekemeijer's dissertation [6].

disturbed by adding Gaussian noise with standard deviation $\sigma = 0.10$ m/s, comparable to current lidar standards [24].[6]

The observers were tuned according to the root mean square (RMS) error, the variance accounted for (VAF), and the quality of fit (QOF) for the mean wake centerline. The mean wake centerline is defined as the longitudinal wind speeds throughout the domain averaged in lateral direction from rotor end to rotor end. The performance measures are defined as

$$\text{RMS}(\overline{u}, \overline{u}_{\text{est}}) = \sqrt{\text{mean}\left((\overline{u} - \overline{u}_{\text{est}})^2\right)}, \tag{4}$$

$$\text{VAF}(\overline{u}, \overline{u}_{\text{est}}) = \left(1 - \frac{\text{variance}(\overline{u} - \overline{u}_{\text{est}})}{\text{variance}(\overline{u})}\right) \cdot 100\%, \tag{5}$$

$$\text{QOF}(\overline{u}, \overline{u}_{\text{est}}) = \max\left(\left[0, 1 - \frac{||\overline{u} - \overline{u}_{\text{est}}||_2^2}{||\overline{u}||_2^2}\right]\right) \cdot 100\%, \tag{6}$$

in which we want to minimize the RMS error, and maximize the VAF and QOF. Tuning the observers resulted in the settings as shown in table 2, with $n_e$ the number of ensembles in the EnKF, $P_0$ the initial state covariance matrix, $R$ the measurement noise covariance matrix and $Q$ the process noise covariance matrix.

Table 2: Optimal observer settings for WFSim at 25x50 meshing.

| **Filter** | **z (-)** | **$\mathbf{n_e}$ (-)** | **$\mathbf{P_0}$ (m/s)** | **R (m/s)** | **Q (m/s)** |
|---|---|---|---|---|---|
| ApkF | 0.05 | N.A. | 10.0 | 1.0 | 0.050 |
| EnKF | N.A. | 200 | 5.0 | 0.10 | 1.00 |

Simulating WFSim without observer, and with the ApKF and EnKF respectively, yields estimates of the flow field at every time instant. The mean wake centerlines for the entire simulation time can be summarized in a single figure, as displayed in figure 7. In this figure, the black line indicates the changes in blade pitch angle of turbine 1, causing flow excitation. Gaps indicate a pitch angle of 4°, and the black lines indicate an angle of 0°.

Figure 7 shows that WFSim predicts an averaged flow, as expected since WFSim does not include sophisticated turbulence or rotor models. The observers attempt to further carve out dynamics seen in SOWFA, and improve wake recovery modeling. Quantitative results (RMS error, VAF and QOF) are displayed in table 3, showing that the observers significantly improve estimations with respect to the validation case. Furthermore, the EnKF slightly outperforms the ApKF, expected to be because the EnKF deals better with nonlinear dynamics in general.

Secondly, the full flow fields are looked at. For time $t = 200, 750$ and 1500 s, the flow fields are presented in figure 8. From these figures, it is further consolidated that WFSim predicts an averaged flow, and neglects smaller scale dynamics, showing satisfactory results in general. The observers correct for a number of factors that WFSim estimates poorly. First of all, it is seen that the wake width is adjusted for: WFSim overestimates the wake width, expected to be due to the very simplified turbulence model. Secondly, in SOWFA, there is no wake formation behind the turbine hub, and thus two wakes form in parallel instead. Due to the simplified actuator disk model in WFSim, a single wake is modeled. The observers correct for this. Thirdly, wake meandering is somewhat accounted for. Fourthly, wake recovery is improved by the observers.

[6] Note that this is not a very realistic number or location of measurements. Reducing the number of measurements decreases performance, but still is beneficial when compared to the case in absence of an observer. For demonstration purposes, more measurements are fed into the observer in these simulations.

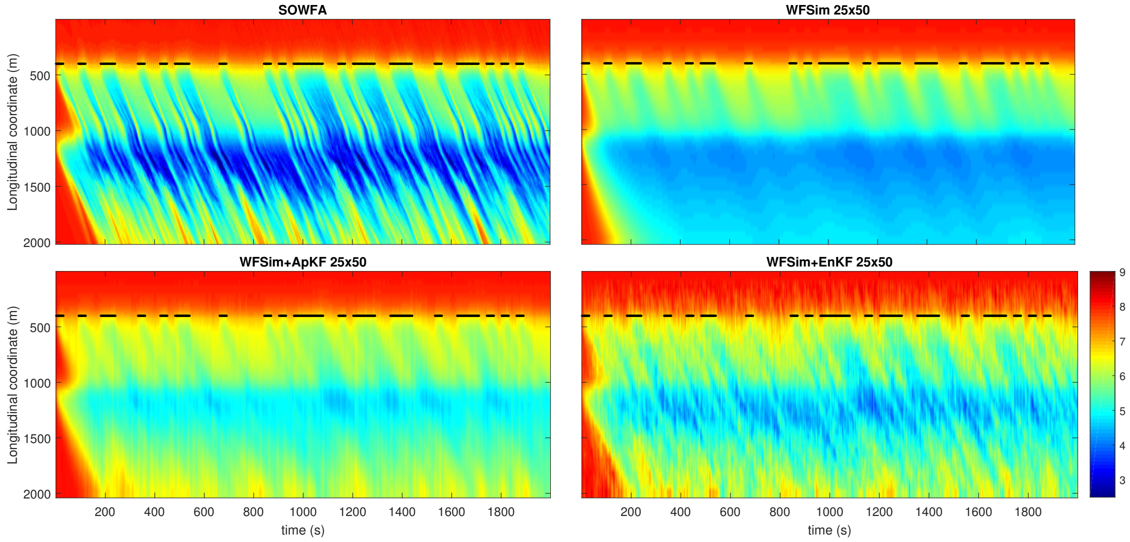


Figure 7: Mean wake centerline velocity in longitudinal direction over the simulation time. The black line indicates the blade pitch angle setting of turbine 1. For the gaps, the blade pitch angle is 4°, and for the black lines it is 0°.

Table 3: RMS, VAF and QOF for the mean wake centerline for WFSim, EnKF and ApKF.

| | **RMS (m/s)** | | | **VAF (%)** | | | **QOF (%)** | | |
|---|---|---|---|---|---|---|---|---|---|
| **Time (s)** | **WFSim** | **EnKF** | **ApKF** | **WFSim** | **EnKF** | **ApKF** | **WFSim** | **EnKF** | **ApKF** |
| 1 | 0.046 | 0.046 | 0.046 | 17.2 | 17.2 | 17.0 | 100 | 100 | 100 |
| 200 | 0.491 | 0.374 | 0.419 | 85.3 | 88.1 | 88.8 | 99.4 | 99.5 | 99.6 |
| 500 | 0.836 | 0.438 | 0.590 | 64.6 | 88.5 | 82.6 | 98.1 | 99.1 | 99.5 |
| 1000 | 0.642 | 0.367 | 0.433 | 77.9 | 90.7 | 88.1 | 98.8 | 99.5 | 99.6 |
| 1500 | 0.785 | 0.415 | 0.448 | 68.8 | 89.2 | 88.8 | 98.3 | 99.4 | 99.5 |
| 1999 | 0.499 | 0.315 | 0.414 | 88.6 | 93.5 | 93.0 | 99.3 | 99.5 | 99.7 |

Finally, comparing the EnKF to the ApKF, it is directly seen that the EnKF relies more on stochasticity, but therefore also deals better with model nonlinearity. Noticeably, the EnKF deals well with the underestimated wake depth, while the ApKF does so less well.

Moreover, the computational cost of observer updates are displayed in table 4, for our simulations with $N = 1034$ and for a larger scale simulation $N = 4560$. This table clearly shows the increase in efficiency in the enhanced filtering algorithms with respect to the traditional KF. Furthermore, when comparing results for $N = 1034$ and $N = 4560$, it is noted that gains in computational efficiency scale up more than linearly with system size.

Table 4: Average computational cost per iteration in seconds.

| **System size** | **WFSim** | **KF** | **EnKF** | **ApKF** |
|---|---|---|---|---|
| $N = 1034$ | $1.0 \cdot 10^{-1}$ | $1.3 \cdot 10^{1}$ | $7.0 \cdot 10^{-1}$ | $8.0 \cdot 10^{-1}$ |
| $N = 4560$ | $1.2 \cdot 10^{0}$ | $1.1 \cdot 10^{3}$ | $8.0 \cdot 10^{0}$ | $8.5 \cdot 10^{0}$ |

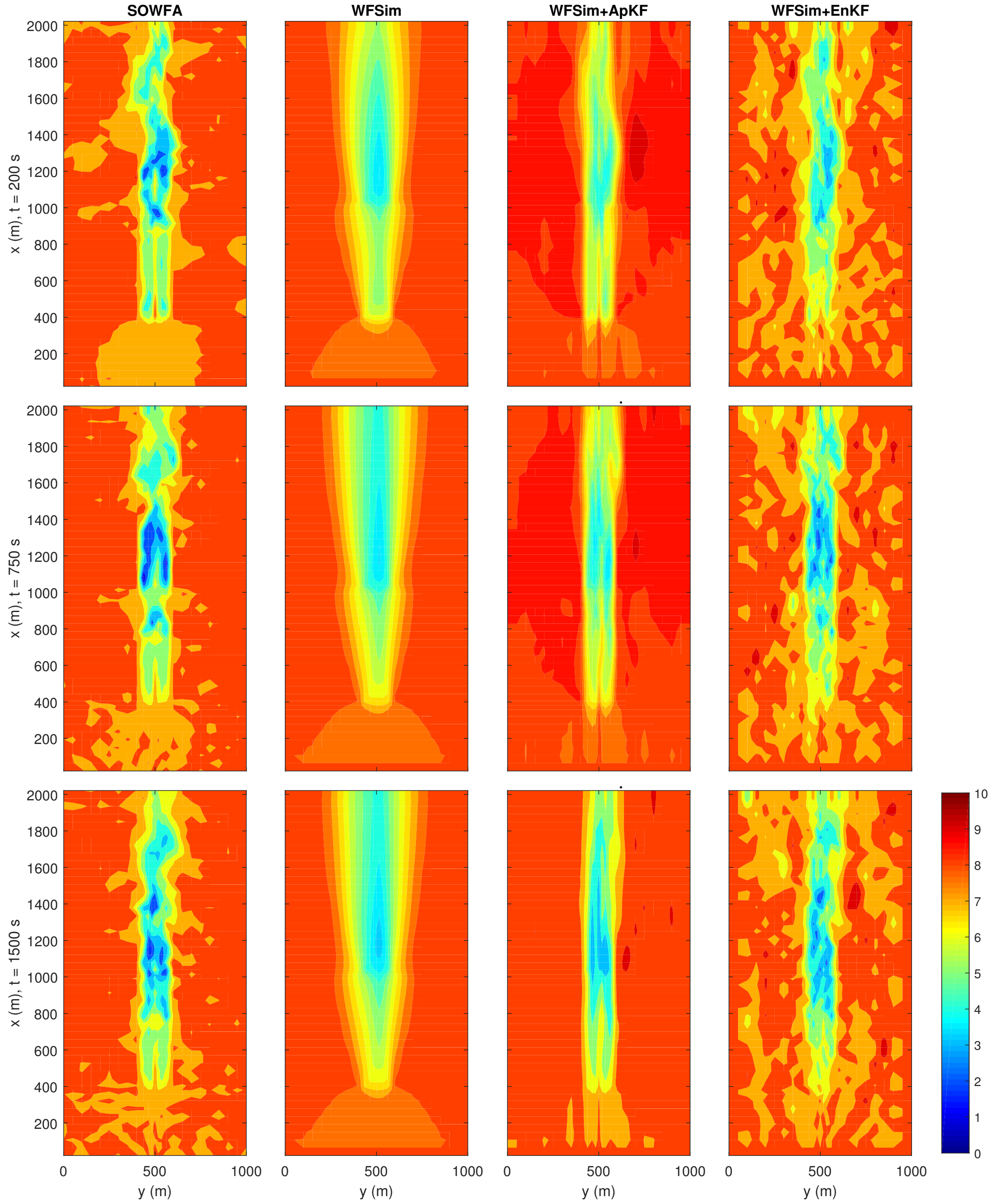


Figure 8: Estimated flow field in x-direction for $t = 200, 750, 1500$ s.

## 6. Conclusions and on-going work

Concluding from figures 7 and 8 and tables 3 and 4, two state observer designs have been proposed with comparable or better performance than a full KF at a fraction of the computation cost. These computational benefits grow more than linearly with system size. In our simulations with $N = 1034$ states, the observers account for an overestimated wake width, underestimated wake depth, underestimated wake recovery, turbine hub effects and wake meandering. This is a first major step to time-efficient higher fidelity closed-loop control of wind farms.

Improvements in model validation and filtering results can be realized by improving the WFSim model (e.g., including turbine hub effects, including a mapping from physical turbine settings to axial induction factors, improving the turbulence model, optimizing the number and location of measurements fed into the observers). Also, more refined tuning of observers (noteably, the covariance matrices and underlying filter assumptions) and domain meshing should further improve state filtering results. Finally, as the focus in this work is limited to axial-induction-based control, wake redirection control is to be explored in future research.

### Acknowledgments

This work would not have been possible without the support of the National Renewable Energy Laboratory (NREL) and the University of Colorado Boulder (CU Boulder).

### References

[1] Goit J and Meyers J 2015 *Journal of Fluid Mechanics* **768** 5–50

[2] Fleming P A, Ning A, Gebraad P M O and Dykes K 2015 *Wind Energy* **19**(2) 329–344

[3] Gebraad P M O 2014 *Data-driven wind plant control* Ph.D. dissertation Delft University of Technology

[4] Fleming P A, Aho J, Gebraad P M O, Pao L Y and Zhang Y 2016 *American Control Conference*

[5] Ela E *et al.* 2014 Active power controls from wind power: Bridging the gaps Technical report National Renewable Energy Laboratory

[6] Doekemeijer B M 2016 *Enhanced Kalman filtering for the 2D CFD Navier-Stokes wind farm model* MSc dissertation

[7] Boersma S, van Wingerden J W, Vali M and Kühn M 2016 *American Control Conference*

[8] Torres P, van Wingerden J W and Verhaegen M 2011 *Control and Automation* **9** 963–968

[9] Versteeg H K and Malalasekera W 2007 *An introduction to Computational Fluid Dynamics: The Finite Volume Method* 2nd ed (Pearson Education Limited) ISBN 978-0-13-127498-3

[10] Bianchi F D, de Battista H and Mantz R J 2007 *Wind Turbine Control Systems: Principles, Modeling and Gain Scheduling Design* 1st ed (Springer) ISBN 9781846284922

[11] Prandtl L 1925 *Journal of Applied Mathematics and Mechanics* **5** 136–139

[12] Ishihara J Y, Terra M H and Bianco A F 2010 *Automatica* **46**(4) 761–766

[13] Ishihara J Y and Terra M H 2008 *Automatica* **44**(8) 2185–2190

[14] Cuthill E and McKee J 1969 Reducing the bandwidth of sparse symmetric matrices *Proceedings of the 1969 24th National Conference* ACM '69 (New York, NY, USA: ACM) pp 157–172

[15] George A and Liu J W 1981 *Computer Solution of Large Sparse Positive Definite* (Prentice Hall) ISBN 0131652745

[16] Evensen G 2003 *Ocean Dynamics* **53**(4) 343–367

[17] Fleming P A, Gebraad P M O, Lee S, van Wingerden J W, Johnson K, Churchfield M, Michalakes J, Spalart P and Moriarty P 2013 High-fidelity simulation comparison of wake mitigation control strategies for a two-turbine case *International Conference on Aerodynamics of Offshore Wind Energy Systems and Wakes*

[18] Annoni J, Seiler P, Johnson K, Fleming P and Gebraad P 2014 *American Control Conference* 2517–2523 ISSN 0743-1619

[19] Jonkman J M and Buhl M L 2005 FAST v6.0 user's guide Technical report National Renewable Energy Laboratory

[20] Fleming P A, Gebraad P M O, Lee S, van Wingerden J W, Johnson K, Churchfield M, Michalakes J, Spalart P and Moriarty P 2014 *Renewable Energy* **70** 211–218 ISSN 0960-1481

[21] Fleming P A, Gebraad P M O, van Wingerden J W, Lee S, Churchfield M, Scholbrock A, Michalakes J, Johnson K and Moriarty P 2013 SOWFA Super-Controller: A high fidelity tool for evaluating wind plant control approaches

[22] Churchfield M J, Lee S, Michalakes J and Moriarty P J 2012 *Journal of Turbulence* **13** 1–32 ISSN 1468-5248

[23] Jonkman J, Butterfield S, Musial W and Scott G 2009 Definition of a 5-MW reference wind turbine for offshore system development Technical report National Renewable Energy Laboratory

[24] Courtney M, Wagner R and Lindelöw P 2008 Commercial lidar profilers for wind energy: a comparative guide *Conference Proceedings of EWEC*

# Bibliography

[1] B. M. Doekemeijer, “Modeling and control of wind farms: a literature survey,” literature survey, Delft University of Technology, November 2015.

[2] International Energy Agency (IEA), “World Energy Outlook 2015: Fact sheet,” 2015 (Accessed: 02-27-2016). http://www.worldenergyoutlook.org/media/weowebsite/2014/141112_WEO_FactSheets.pdf.

[3] C.-K. Park, H.-R. Byun, R. Deo, and B.-R. Lee, “Drought prediction till 2100 under RCP 8.5 climate change scenarios for Korea,” *Journal of Hydrology*, vol. 526, pp. 221 – 230, 2015.

[4] S. Vardoulakis, C. Dimitroulopoulou, J. Thornes, K.-M. Lai, J. Taylor, I. Myers, C. Heaviside, A. Mavrogianni, C. Shrubsole, Z. Chalabi, M. Davies, and P. Wilkinson, “Impact of climate change on the domestic indoor environment and associated health risks in the UK,” *Environment International*, vol. 85, pp. 299 – 313, 2015.

[5] W. Haeberli, C. Huggel, F. Paul, and M. Zemp, “Glacial responses to climate change,” in *Treatise on Geomorphology* (J. F. Shroder, ed.), ch. 13, pp. 152 – 175, San Diego: Academic Press, 2013.

[6] B. Worm and H. K. Lotze, “Marine biodiversity and climate change,” in *Climate Change* (T. M. Letcher, ed.), ch. 13, pp. 195 – 212, Boston: Elsevier, 2nd ed., 2016.

[7] S. Shafiee and E. Topal, “When will fossil fuel reserves be diminished,” *Energy Policy*, vol. 37, no. 1, pp. 181–189, 2009.

[8] K. M. Hovland, “Denmark’s wind power output rises to record in first half,” *The Wall Street Journal*, September 2014. http://www.wsj.com/articles/denmarks-wind-power-output-rises-to-record-in-first-half-1409750563.

[9] J. Goit and J. Meyers, “Optimal control of energy extraction in wind-farm boundary layers,” *Journal of Fluid Mechanics*, vol. 768, pp. 5–50, 2015.

[10] C. Steiness, "Photograph: Wake formation in the Horns Rev offshore wind farm near Denmark," 2008. http://www.popsci.com/sites/popsci.com/files/styles/large_1x_/public/import/2013/images/2010/01/horns_rev.jpg?itok=NiFLyV42.

[11] L. Y. Pao and K. E. Johnson, "A tutorial on the dynamics and control of wind turbines and wind farms," *American Control Conference*, pp. 2076–2089, 2009.

[12] P. M. O. Gebraad, *Data-driven wind plant control.* Ph.D. dissertation, Delft University of Technology, 2014. Electronic copy available at http://repository.tudelft.nl.

[13] E. Ela, V. Gevorgian, P. Fleming, Y. C. Zhang, M. Singh, E. Muljadi, A. Scholbrock, J. Aho, A. Buckspan, L. Y. Pao, V. Singhvi, A. Tuohy, P. Pourbeik, D. Brooks, and N. Bhatt, "Active power controls from wind power: Bridging the gaps," technical report, National Renewable Energy Laboratory (NREL), 2014.

[14] T. Knudsen, T. Bak, and M. Svenstrup, "Survey of wind farm control-power and fatigue optimization," *Wind Energy*, vol. 18, no. 8, pp. 1333–1351, 2015.

[15] P. A. Fleming, P. M. O. Gebraad, S. Lee, J. W. van Wingerden, K. Johnson, M. Churchfield, J. Michalakes, P. Spalart, and P. Moriarty, "High-fidelity simulation comparison of wake mitigation control strategies for a two-turbine case," in *International Conference on Aerodynamics of Offshore Wind Energy Systems and Wakes (ICOWES)*, June 2013.

[16] P. A. Fleming, P. M. O. Gebraad, S. Lee, J. W. van Wingerden, K. Johnson, M. Churchfield, J. Michalakes, P. Spalart, and P. Moriarty, "Evaluating techniques for redirecting turbine wakes using SOWFA," *Renewable Energy*, vol. 70, pp. 211–218, 2014.

[17] D. G. Wilson, D. E. Berg, B. R. Resor, M. F. Barone, and J. C. Berg, "Combined individual pitch control and active aerodynamic load controller investigation for the 5MW upwind turbine," in *American Wind Energy Association (AWEA)*, May 2013.

[18] E. A. Bossanyi, "Individual blade pitch control for load reduction," *Wind Energy*, vol. 6, no. 2, pp. 119–128, 2003.

[19] E. A. Bossanyi, "Further load reductions with individual pitch control," *Wind Energy*, vol. 8, no. 4, pp. 481–485, 2005.

[20] P. A. Fleming, P. M. O. Gebraad, J. W. van Wingerden, S. Lee, M. Churchfield, A. Scholbrock, J. Michalakes, K. Johnson, and P. Moriarty, "SOWFA Super-Controller: A high fidelity tool for evaluating wind plant control approaches," February 2013.

[21] B. Sanderse, S. P. van der Pijl, and B. Koren, "Review of computational fluid dynamics for wind turbine wake aerodynamics," *Wind Energy*, vol. 14, no. 7, pp. 799–819, 2011.

[22] F. D. Bianchi, H. de Battista, and R. J. Mantz, *Wind Turbine Control Systems: Principles, Modeling and Gain Scheduling Design.* Springer, 1st ed., 2007.

[23] J. M. Jonkman and M. L. Buhl, "FAST v6.0 user's guide," technical report, National Renewable Energy Laboratory (NREL), August 2005.

[24] N. O. Jensen, "A note on wind generator interaction," technical report, Risø National Laboratory, 1983.

[25] J. F. Ainslie, "Calculating the flowfield in the wake of wind turbines," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 27, no. 1-3, pp. 213–224, 1988.

[26] P. Fleming, P. Gebraad, M. Churchfield, S. Lee, K. Johnson, J. Michalakes, J.-W. van Wingerden, and P. Moriarty, "SOWFA + Super Controller user's manual," technical report, National Renewable Energy Laboratory (NREL), August 2013.

[27] P. Torres, J. W. van Wingerden, and M. Verhaegen, "Modeling of the flow in wind farms for total power optimization," *Control and Automation*, vol. 9, pp. 963–968, 2011.

[28] S. Boersma, J. W. van Wingerden, M. Vali, and M. Kühn, "Quasi linear parameter varying modeling for wind farm control using the 2D Navier Stokes equations," *American Control Conference*, 2016.

[29] S. Boersma and J. W. van Wingerden, "A control-oriented dynamic flow model: "WF-Sim"," *Journal of Physics: Conference Series*, 2016.

[30] M. Vali, J. W. van Wingerden, S. Boersma, and M. Kühn, "A predictive control benchmark for optimal energy extraction of wind farms," *Journal of Physics: Conference Series*, 2016.

[31] H. K. Versteeg and W. Malalasekera, *An introduction to Computational Fluid Dynamics: The Finite Volume Method.* Pearson Education Limited, 2nd ed., 2007.

[32] L. Prandtl, "Uber die ausgebildete turbulenz," *Journal of Applied Mathematics and Mechanics*, vol. 5, pp. 136–139, 1925.

[33] P. M. O. Gebraad, "The mixing length turbulence model in WindFarmSimulator 2.2." Unpublished, March 2012.

[34] S. Boersma, *WindFarmSimulator: a dynamic medium fidelity wind farm model.* Ph.D. dissertation, Delft University of Technology. In preparation.

[35] G. C. Larsen, H. A. Madsen, F. Bingöl, J. Mann, S. Ott, J. N. Sørensen, V. Okulov, N. Troldborg, M. Nielsen, K. Thomsen, T. J. Larsen, and R. Mikkelsen, "Dynamic wake meandering modelling," technical report, Risø National Laboratory, June 2007.

[36] M. J. Churchfield, S. Lee, J. Michalakes, and P. J. Moriarty, "A numerical study of the effects of atmospheric and wake turbulence on wind turbine dynamics," *Journal of Turbulence*, vol. 13, no. 14, pp. 1–32, 2012.

[37] J. Annoni, P. Seiler, K. Johnson, P. Fleming, and P. Gebraad, "Evaluating wake models for wind farm control," *American Control Conference*, pp. 2517–2523, 2014.

[38] P. A. Fleming, A. Ning, P. M. O. Gebraad, and K. Dykes, "Wind plant system engineering through optimization of layout and yaw control," *Wind Energy*, 2015.

[39] J. Jonkman, S. Butterfield, W. Musial, and G. Scott, "Definition of a 5-MW reference wind turbine for offshore system development," technical report, National Renewable Energy Laboratory (NREL), February 2009.

[40] P. Gebraad, J. J. Thomas, A. Ning, P. Fleming, and K. Dykes, "Maximization of the annual energy production of wind power plants by optimization of layout and yaw-based wake control," 2015. (in review).

[41] P. Gebraad, "SOWFA data en FAST data." Private email conversation, November 2015.

[42] J. Annoni and P. Seiler, "A low-order model for wind farm control," in *AMCC*, 2015.

[43] M. Verhaegen and V. Verdult, *Filtering and System Identification: A Least Squares Approach.* Cambridge University Press, 1st ed., 2007.

[44] J. Y. Ishihara and M. H. Terra, "Robust state prediction for descriptor systems," *Automatica*, vol. 44, pp. 2185–2190, 2008.

[45] J. Y. Ishihara, M. H. Terra, and A. F. Bianco, "Recursive linear estimation for general discrete-time descriptor systems," *Automatica*, vol. 46, pp. 761–766, 2010.

[46] E. Cuthill and J. McKee, "Reducing the bandwidth of sparse symmetric matrices," in *Proceedings of the 1969 24th National Conference*, ACM '69, pp. 157–172, ACM, 1969.

[47] A. George and J. W. Liu, *Computer Solution of Large Sparse Positive Definite.* Prentice Hall, 1981.

[48] C. Poussot-Vassal and D. Sipp, "Parametric reduced order dynamical model construction of a fluid flow control problem," *International Federation of Automatic Control*, vol. 48, pp. 133–138, October 2015. 1st Workshop on Linear Parameter Varying Systems.

[49] G. Evensen, "The ensemble kalman filter: theoretical formulation and practical implementation," *Ocean Dynamics*, vol. 53, pp. 343–367, 2003.

[50] S. Gillijns, O. B. Mendoza, J. Chandrasekar, B. L. R. D. Moor, D. S. Bernstein, and A. Ridley, "What is the ensemble kalman filter and how well does it work?," in *American Control Conference*, pp. 4448–4453, June 2006.

[51] E. Ott, B. R. Hunt, I. Szunyogh, A. V. Zimin, E. J. Kostelich, M. Corazza, E. Kalnay, D. Patil, and J. A. Yorke, "A local ensemble kalman filter for atmospheric data assimilation," *Tellus A*, vol. 56, no. 5, pp. 415–428, 2004.

[52] R. H. Reichle, D. B. McLaughlin, and D. Entekhabi, "Hydrologic data assimilation with the ensemble kalman filter," *Monthly Weather Review*, vol. 130, no. 1, pp. 103–114, 2002.

[53] P. L. Houtekamer, H. L. Mitchell, G. Pellerin, M. Buehner, M. Charron, L. Spacek, and B. Hansen, "Atmospheric data assimilation with an ensemble kalman filter: Results with real observations," *Monthly weather review*, vol. 133, no. 3, pp. 604–620, 2005.

[54] P. L. Houtekamer and H. L. Mitchell, "Ensemble kalman filtering," *Quarterly Journal of the Royal Meteorological Society*, vol. 131, no. 613, pp. 3269–3289, 2005.

[55] D. B. Work, O. P. Tossavainen, S. Blandin, A. M. Bayen, T. Iwuchukwu, and K. Tracton, "An ensemble kalman filtering approach to highway traffic estimation using gps enabled mobile devices," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pp. 5062–5068, December 2008.

[56] J. Xu, Y. Zhou, and Y. Jing, “Extended target tracking for high resolution sensor based ensemble kalman filters,” in *Control and Decision Conference (CCDC), 2010 Chinese*, pp. 3308–3313, May 2010.

[57] M. Courtney, R. Wagner, and P. Lindelöw, “Commercial lidar profilers for wind energy: a comparative guide,” in *Conference Proceedings of EWEC*, 2008.

[58] W. Tong, *Wind Power Generation and Wind Turbine Design.* WIT Press, 1st ed., April 2010.

[59] R. E. Keck and O. Undheim, “A pragmatic approach to wind farm simulations using the dynamic wake meandering model,” *Wind Energy*, vol. 18, no. 9, pp. 1671–1682, 2015.

[60] I. Paraschivoiu, *Wind Turbine Design: Emphasis on Darrieus Concept.* Ecole Polytechnique de Montreal, 1st ed., 2002.

# Glossary

## List of Acronyms

| | |
|---|---|
| **ADM** | Actuator disk model |
| **ALM** | Actuator line model |
| **ApKF** | Approximate Kalman filter |
| **CFD** | Computational fluid dynamics |
| **CPU** | Central processing unit |
| **CU** | University of Colorado Boulder |
| **DCSC** | Delft Center for Systems and Control |
| **DOE** | Department of Energy |
| **DTU** | Technical University of Denmark |
| **DWM** | Dynamic Wake Meandering |
| **ECEE** | Electrical, Computer and Energy Engineering |
| **EnKF** | Ensemble Kalman filter |
| **FAST** | Fatigue, Aerodynamics, Structures and Turbulence |
| **FLORIS** | FLOw Redirection and Induction in Steady-state |
| **FLORIDyn** | FLOw Redirection and Induction Dynamics |
| **IEA** | International Energy Agency |
| **IPC** | Individual pitch control |
| **IRKA** | Iterative Tangential Interpolation Algorithm |
| **KF** | Kalman filter |

| | |
|---|---|
| **LCOE** | Levelized cost of energy |
| **LES** | Large-eddy simulation |
| **lidar** | Light detection and ranging |
| **LPF** | Low-pass filter |
| **LTI** | Linear time-invariant |
| **NREL** | National Renewable Energy Laboratory |
| **NS** | Navier-Stokes |
| **PDE** | Partial differential equation |
| **PRBS** | Pseudo-random binary sequence |
| **qLPV** | Quasi-linear parameter varying |
| **QOF** | Quality of fit |
| **RANS** | Reynolds Averaging of Navier-Stokes |
| **RCM** | Reverse Cuthill-McKee |
| **RMS** | Root mean square |
| **SOWFA** | Simulator fOr Wind Farm Applications |
| **SVD** | Singular value decomposition |
| **TORQUE** | Science of Making Torque from Wind |
| **TSR** | Tip-speed-ratio |
| **TU Delft** | Delft University of Technology |
| **VAF** | Variance accounted for |
| **WESys** | Wind Energy System |
| **WFSim** | WindFarmSimulator |

## List of Symbols

| | | |
|---|---|---|
| $\alpha$ | System state vector for WindFarmSimulator | - |
| $\alpha_t$ | Tuning parameter in the wake centerline curve | - |
| $\tilde{\alpha}$ | Extended system state vector of WindFarmSimulator | - |
| $\hat{\tilde{\alpha}}$ | Filtered/predicted state estimate | - |
| $\beta$ | Rotor blade pitch angle | $^\circ$ |
| $\beta_w$ | System input vector for WindFarmSimulator | - |
| $\gamma$ | Turbine yaw angle | $^\circ$ |

| | | |
|---|---|---|
| $\gamma_e$ | Measurement noise ensemble | m/s |
| $\Delta_{i-j}$ | Unit pulse nonzero only for $i = j$ | - |
| $\epsilon^m$ | Measurement noise | m/s |
| $\epsilon^p$ | Process noise | - |
| $\lambda$ | Tip-speed-ratio | - |
| $\Lambda$ | Eigenvalues matrix obtained by squaring the SVD diagonal values matrix $\Sigma$ | - |
| $\mu$ | Dynamic viscosity | $\mathrm{N} \cdot \mathrm{s} \cdot \mathrm{m}^{-2}$ |
| $\rho$ | Flow (atmospheric) density | $\mathrm{N} \cdot \mathrm{s}^2 \cdot \mathrm{m}^{-4}$ |
| $\sigma$ | Standard deviation | - |
| $\Sigma$ | Matrix of SVD containing the diagonal values | - |
| $\tau_g$ | Generator torque | $\mathrm{N} \cdot \mathrm{m}$ |
| $\tau_{uu}$ | Longitudinal Reynolds normal stress | Pa |
| $\tau_{uv}$ | Reynolds shear stress | Pa |
| $\tau_{vu}$ | Reynolds shear stress | Pa |
| $\tau_{vv}$ | Lateral Reynolds normal stress | Pa |
| $\psi$ | Model forecast or model analysis as used in the EnKF | - |
| $\omega$ | Rotor rotational speed | $\mathrm{rad} \cdot \mathrm{s}^{-1}$ |
| | | |
| $a$ | Axial induction factor | - |
| $A$ | Cross-sectional area of a flow | $\mathrm{m}^2$ |
| $A_\infty$ | Cross-sectional area of a freestream flow | $\mathrm{m}^2$ |
| $A_{-\infty}$ | Cross-sectional area of a flow far downstream | $\mathrm{m}^2$ |
| $A_D$ | Cross-sectional area of the infinitely thin disk (or: rotor swept area) | $\mathrm{m}^2$ |
| $A_s$ | System matrix for an LTI system | - |
| $A_e$ | System state vector ensemble | - |
| $A_w$ | System matrix for WindFarmSimulator | - |
| $A_x$ | Matrix in discretized NS equality of WFSim originating from convection and difussion terms of the $u-$ and $v-$momentum equations | - |
| $A_y$ | Matrix in discretized NS equality of WFSim originating from convection and difussion terms of the $u-$ and $v-$momentum equations | - |
| $A'_e$ | Deviation in state ensemble from the mean | - |
| $\overline{A_e}$ | State ensemble mean | - |
| $\hat{A}$ | System matrix dependent on the Reynolds number | - |
| $b_x$ | Vector in discretized NS equality of WFSim originating from the mesh layout and velocity vectors at a previous time instant | - |
| $b_y$ | Vector in discretized NS equality of WFSim originating from the mesh layout and velocity vectors at a previous time instant | - |
| $b_c$ | Vector in discretized NS equality of WFSim originating from the boundary conditions | - |
| $B_p$ | System matrix for WindFarmSimulator used for projecting away the continuity equation, obtained as a solution to the continuity equation | - |
| $B_s$ | System matrix for an LTI system | - |
| $B_w$ | State-dependent system matrix for WindFarmSimulator | - |
| $B_x$ | Matrix in discretized NS equality of WFSim originating from the mesh layout | - |
| $B_y$ | Matrix in discretized NS equality of WFSim originating from the mesh layout | - |
| $\tilde{B}$ | Manipulated system matrix of WindFarmSimulator | - |
| $\hat{B}$ | System matrix dependent on the Reynolds number | - |
| $C_P$ | Non-dimensional power coefficient | - |
| $C_s$ | System matrix for an LTI system | - |
| $C_T$ | Non-dimensional trust coefficient | - |
| $\tilde{C}$ | Manipulated system matrix of WindFarmSimulator | - |
| $\hat{C}$ | System matrix dependent on the Reynolds number | - |

| | | |
|---|---|---|
| $d$ | Measurement vector | m/s |
| $dt$ | Timestep used in WFSim | s |
| $D$ | Rotor diameter | m |
| $D_e$ | Measurement ensemble | m/s |
| $D_s$ | System matrix for an LTI system | - |
| $D'_e$ | Deviation in measurement ensemble from the mean | - |
| $e_k$ | State estimation error | - |
| $E$ | Expected value operator (from probability theory) | - |
| $E_w$ | State-dependent system matrix for WindFarmSimulator | - |
| $\hat{E}$ | System matrix dependent on the Reynolds number | - |
| $\tilde{F}$ | Manipulated system matrix of WindFarmSimulator | - |
| $F_D$ | Axial force exerted on the flow by the infinitely thin disk | N |
| $F_T$ | Axial force exerted on the flow by a turbine at hub height | N |
| $h$ | Timestep as defined by user | s |
| $h_z$ | Turbine hub height | m |
| $H$ | Approximated system output matrix in the EnKF | - |
| $i$ | Indexing integer | - |
| $I$ | Indexing integer | - |
| $j$ | Indexing integer | - |
| $J$ | Indexing integer | - |
| $k$ | Discrete time index | - |
| $l_m$ | Eddy length scale | m |
| $L_k$ | Gain matrix for predicted state estimate update in KF | - |
| $L_w$ | Wake half width | m |
| $L_x$ | Longitudinal mesh length (or: rectangular domain length) | m |
| $L_y$ | Lateral mesh length (or: rectangular domain width) | m |
| $L'_k$ | Gain matrix for filtered state estimate update in KF | - |
| $L^\star$ | Steady state optimal gain matrix for the regular KF | - |
| $M$ | Size of system output vector | - |
| $M_z$ | Tower fore-aft bending moment | $\text{N} \cdot \text{m}$ |
| $n$ | Gearbox ratio | - |
| $n_e$ | Number of ensemble members | - |
| $n_x$ | Unit vector in $x$-direction | - |
| $n_y$ | Unit vector in $y$-direction | - |
| $N$ | Size of system state vector | - |
| $NN$ | Total number of steps in a simulation | - |
| $N_x$ | Number of longitudinal mesh points | - |
| $N_y$ | Number of lateral mesh points | - |
| $O$ | Size of system input vector | - |
| $p_0$ | Initial pressure in the flow field | Pa |
| $p$ | Pressure in a flow field | $\text{N} \cdot \text{m}^{-2}$ |
| $\overline{p}$ | Vector containing all pressure terms in a specified mesh | Pa |
| $P$ | Power captured by a wind turbine | W |
| $P_e$ | Sample state estimation error covariance matrix | - |
| $P_k$ | State estimation error covariance matrix at time $k$ | - |
| $P_{\text{rated}}$ | Rated power captured by a wind turbine | W - |
| $P_V$ | Power available in a freestream wind flow | W |
| $P^\star$ | Solution to the discrete algebraic Riccati equation | - |
| $Q_k$ | Process noise covariance matrix at time $k$ | - |
| $Q_p$ | System matrix for WindFarmSimulator used for projecting away the continuity equation | - |
| $R$ | Rotor radius | m |

| | | |
|---|---|---|
| $R_e$ | Sample measurement noise covariance matrix | - |
| $R_k$ | Measurement noise covariance matrix at time $k$ | m/s |
| $Re$ | Reynolds number | - |
| $\overline{S_x}$ | Longitudinal forcing terms in discretized NS equality of WFSim originating from turbine extracting energy from the flow | - |
| $\overline{S_y}$ | Lateral forcing terms in discretized NS equality of WFSim originating from turbine extracting energy from the flow | - |
| $S_k$ | Cross-covariance matrix for measurement and process noise | - |
| $S_{M_u}$ | Longitudinal external force term in NS equations | $\mathrm{N} \cdot \mathrm{m}^{-3}$ |
| $S_{M_v}$ | Lateral external force term in NS equations | $\mathrm{N} \cdot \mathrm{m}^{-3}$ |
| $S_w$ | State-dependent system matrix for WindFarmSimulator | - |
| $t$ | Time | s |
| $T_{M_u}$ | Longitudinal turbulence term | $\mathrm{N} \cdot \mathrm{m}^{-3}$ |
| $T_{M_v}$ | Lateral turbulence term | $\mathrm{N} \cdot \mathrm{m}^{-3}$ |
| $u$ | Longitudinal flow speed | m/s |
| $u_0$ | Vector in discretized NS equality of WFSim originating from convection and difussion terms of the $u-$ and $v-$momentum equations | - |
| $u_k$ | Input vector for an LTI system | - |
| $u'$ | Longitudinal velocity fluctuation from turbulence | m/s |
| $\overline{u}$ | Vector containing all longitudinal velocities in a specified mesh | m/s |
| $\hat{u}$ | System input vector of the Reynolds dependent system | - |
| $U$ | Wind flow velocity | m/s |
| $U_\infty$ | Freestream longitudinal wind speed (far upstream) | m/s |
| $U_{-\infty}$ | Longitudinal wind speed far downstream of a flow | m/s |
| $U_{\text{cut-in}}$ | Cut-in wind speed; wind speed at which control becomes worthwhile | m/s |
| $U_D$ | Flow speed at the infinitely thin disk (or: rotor) | m/s |
| $U_{\text{rated}}$ | Rated wind speed; wind speed after which power generation is limited | m/s |
| $U_{\text{SVD}}$ | Left-side matrix in the SVD | - |
| $v$ | Lateral flow speed | m/s |
| $v_0$ | Vector in discretized NS equality of WFSim originating from convection and difussion terms of the $u-$ and $v-$momentum equations | - |
| $v'$ | Lateral velocity fluctuation from turbulence | m/s |
| $\overline{v}$ | Vector containing all lateral velocities in a specified mesh | m/s |
| $V_\infty$ | Freestream lateral wind speed | m/s |
| $V_{\text{SVD}}$ | Right-side matrix in the SVD | - |
| $x$ | Longitudinal displacement | m |
| $x_k$ | System state vector of an LTI system | - |
| $\hat{x}$ | System state vector of the Reynolds dependent system | - |
| $X_1$ | Longitudinal coordinates for primary WFSim meshing | m |
| $X_2$ | Longitudinal coordinates for secondary WFSim meshing | m |
| $y$ | Lateral displacement | m |
| $y_k$ | System output vector for an LTI system | - |
| $\hat{y}$ | System output vector of the Reynolds dependent system | - |
| $Y_1$ | Lateral coordinates for primary WFSim meshing | m |
| $Y_2$ | Lateral coordinates for secondary WFSim meshing | m |